

Blockchain-Based Energy Accounting System

(23R11A66B4) Dhanush Chintam - Backend

(23R11A66F6) A.S.Varshitha - Frontend

(23R11A66G5) Chitneni Ananya - Frontend

Duration: Summer Internship 2025

Problem Statement

Old energy tracking systems store data in separate places, take too long to update or match records, and make it hard for users to see what's really happening which leads to delays, confusion, and wasted effort.

Why it matters?

With the rise of renewable energy and decentralized grids, there is a growing need for a secure, real-time, and tamper-proof way to track energy generation, usage, and billing.

Solution Overview

Our Approach:

We designed a web-based platform that records energy data from IoT devices, stores it on a blockchain via smart contracts, and provides real-time insights and reports to consumer and producer.



Key Features/Functionality

 Credit System

 Immutable Ledger

 ETH Payments

 P2P Credit Transfers

 Unified User Dashboard

 Intuitive UI Design

Tech Stack:

Frontend: React.js+ Tailwind css

Backend: Node.js + Express

Database: MySQL

Blockchain: Ethereum (Smart Contracts)

Other: Figma/Canva (Design)

Demo

The screenshot displays the EnergyGrid mobile application interface. At the top, there is a navigation bar with the EnergyGrid logo, a bell icon for notifications, and a "My Account" section showing the account address 0x...4096. Below the navigation bar, there are three main buttons: "Transfer Credits" (orange), "pay bill" (teal), and "distribute credits" (pink). Underneath these buttons, there are three summary cards: "Generated" (120 kWh), "Consumed" (50 kWh), and "Credits" (219 EC). A large callout box titled "Recent Transactions" lists five entries with columns for Type, Amount, Status, and Block number.

TYPE	AMOUNT	STATUS	BLOCK
Consumption	50 kWh	Confirmed	21
Generation	120 kWh	Confirmed	20
Generation	120 kWh	Confirmed	17
Consumption	50 kWh	Confirmed	15
Credit Transfer	1 EC	Confirmed	12

Learnings

- Worked with decentralized architecture & blockchain
- Deployed smart contracts and integrated APIs
- Connecting backend to web frontend
- Designed user flows in Figma

Challenges

- Smart contract deployment & gas optimization
- Real-time IoT data syncing
- Visualizing complete user flow

Future Scope

- Plan to develop a web application for convenient access
- Aim to enhance the system for broader scalability
- Layer-2 solutions (Polygon, zk-Rollups)
- Explore opportunities for global integration and adoption

- Project Status: Functional prototype, ready for pilot testing with simulated data.



Thank You