1. Vision: what do you want to do. Do you have an energy goal? Emissions? Costs savings? Improvement of a service?

Reduce household energy consumption and costs by monitoring and controlling electrical loads using smart plugs, real-time data, and user-defined priorities.

- **Energy Goal**: Cost savings and reduced CO_2 emissions.
- **Inspiration**: Similar to banking apps that track spending, this app will track and optimize energy usage.
- **User Benefit**: Empower users to make informed decisions and automate energy-saving actions.

2. Assets: what do you want to measure? What materials do you need? How do you gather data? API vs recorded locally or both? Which sensors do you need?

Sensors & Hardware
- ESP32 microcontroller
- Current ¡sensors
- Smart plugs with relay control
- Relays (5V, 10A) for switching non-critical loads
- Power supply and connectors

Software & Data
- Mobile/Web App (dashboard)
- APIs:
    - ESIOS for electricity prices and CO_2 emissions
    - Weather API (optional)
- Local data storage (CSV or database)
- Telegram Bot for remote control

3. Approach: define a team leader (makes communication with us easier). How are the rest of the roles divided? What is the timeline of your Project? Brainstorm and definition of the communication scheme

Implementation Plan
- **Phase 1**: Hardware setup and sensor calibration
- **Phase 2**: ESP32 programming for data acquisition and relay control
- **Phase 3**: API integration for electricity prices and emissions
- **Phase 4**: App development (dashboard, control interface, scoring system)
- **Phase 5**: Testing, optimization, and user feedback

Team Roles
- **Team Leader**:
- **Hardware Lead**: Sensor and relay integration
- **Software Lead**: ESP32 and app development
- **Data Analyst**: API integration and trend analysis

Communication:
- GitHub for code sharing

4. Outocomes and presentation: How do you measure the success of your Project? How do you present it? (dashboard?)

   Deliverables
- Functional app with:
  - Real-time consumption graphs
  - Comparison with internet data (e.g., electricity prices)
  - Load control based on user priorities
  - Scoring system to incentivize savings
- Dashboard with:
  - Trends and historical data
  - Load prioritization settings
  - Notifications and recommendations
- CSV data logs
- Final presentation with:
  - Live demo
  - Energy savings analysis
  - User experience feedback