Axle Tech Exercise: Engineering

Thanks for your interest in working together. This is meant to be a fun, short exercise which will help you understand what kind of things we might work on together, and to give us a sense of your work. Please let us know if you have feedback - we're always looking to improve our process.

The exercise is in two parts:

1. **Take home component:** A short piece of work, resulting in some code that you send us
2. **Pair programming session:** we will go over your submission together and look for ways to improve it

🐍 We prefer Python solutions, but may be able to review solutions in other languages if you have a strong preference. Please let us know if you plan to use a different language and we'll try to accommodate it. We'll need to run your solution, so please make it clear how to do so in a `README`.

The challenge

In this task, we're building a whitelabel app which our customers can put in the hands of their own end users to manage their electricity usage.

Your challenge is to implement a single screen for this app, namely the charge control panel: a screen the user can use to see what charging we have planned for their electric vehicle, and override it if necessary.

This panel is a little complex, because it needs to:

1. Show the user when the charge is scheduled to happen

2. Allow them to override the schedule and charge immediately if they need to (e.g. they get home and want to go out again in an hour, so they need to charge now)
3. Allow them to stop charging
4. Keep track of whether the current charge is the scheduled one, or the result of an override, in order to determine the behaviour of the stop charging button (see AC's below)

Acceptance criteria

- User should be able to see their battery % (State of Charge or SOC), what the schedule is, and what their battery SoC will be at the end of the scheduled charge.
    - this obviously depends upon the starting battery %, which in reality we might get from the car's API; you can hardcode it to something like 60%
- Users should be able to hit a button to override the schedule and charge the car immediately
    - When they do so, the car should charge for a set period of time (e.g. 60 minutes)
    - At the end of that time, the car should revert to the schedule
    - During this time, if the user hits "Stop Charge", the car should revert to the schedule
- When scheduled charging is occurring, if the user hits "Stop Charge", the schedule should be disabled until the next morning
- The charge schedule should only be applied when the user's car is plugged in - you should include a UI element that highlights whether the car plugged in or not, and makes it clear how this relates to the charging (e.g. if you hit "Start Charge" and you're not plugged in, you're not overriding the schedule: the schedule only applies when the car is plugged in)

Template Streamlit app

- We’ve provided a template app implemented in Streamlit. You can use this as the basis of your submission and just implement the missing functionality.
  - The template app contains a minimal frontend that displays the relevant information.
  - There are `TODO`s in the code that indicate where you need to implement the backend functionality.
- You’re also very welcome to implement your own app from scratch. This can be in a server-rendered app, or have separate back and frontends.

What we’d like you to submit

- Some runnable code, packaged however you want
  - You don’t need to worry about real price data or the actual optimization - please just mock this with something reasonable (e.g. “charge from 2-5am”)
  - Obviously we’re not expecting this to connect to a real car - you should be mocking the car with an object that behaves a bit like a car
  - We don’t mind whether the UI is optimized for desktop or mobile, but it should be intuitive and attractive for end users
- Brief notes on the decisions you made as you went through the task; these could also be as comments in the code
  - what assumptions did you make?
  - which parts did you choose to spend time on; which bits did you consider less important?
  - how you designed the system bearing in mind its end use.