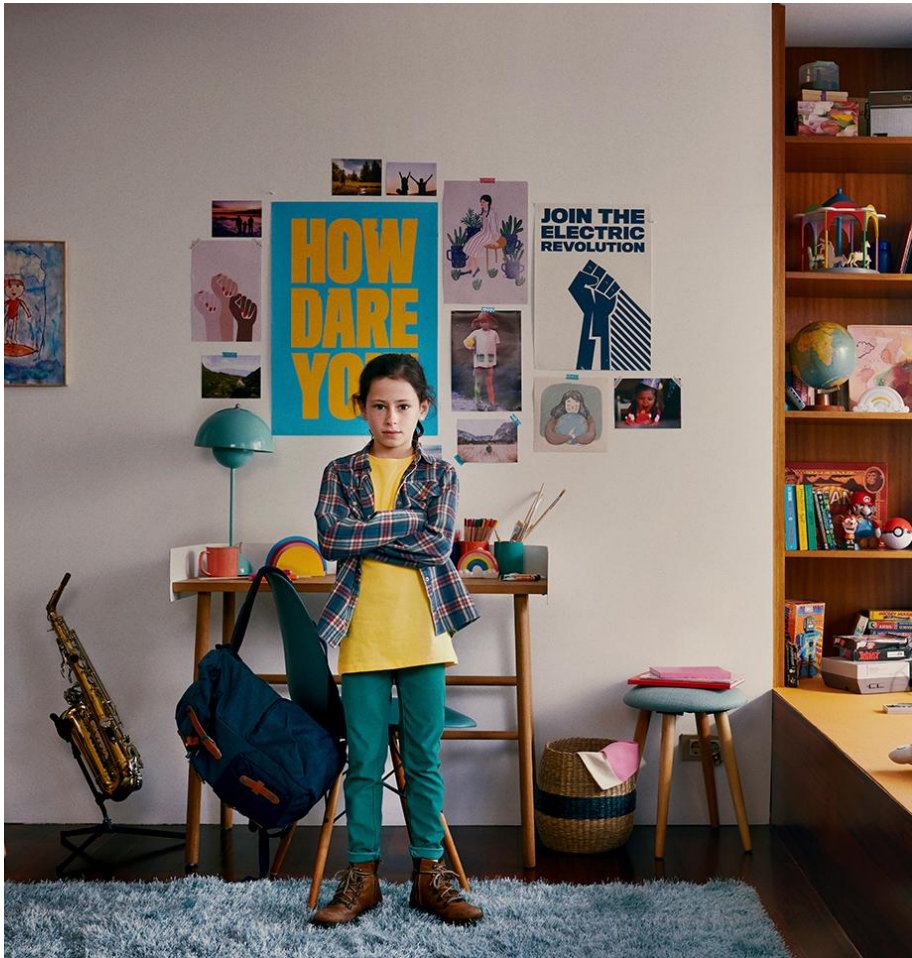
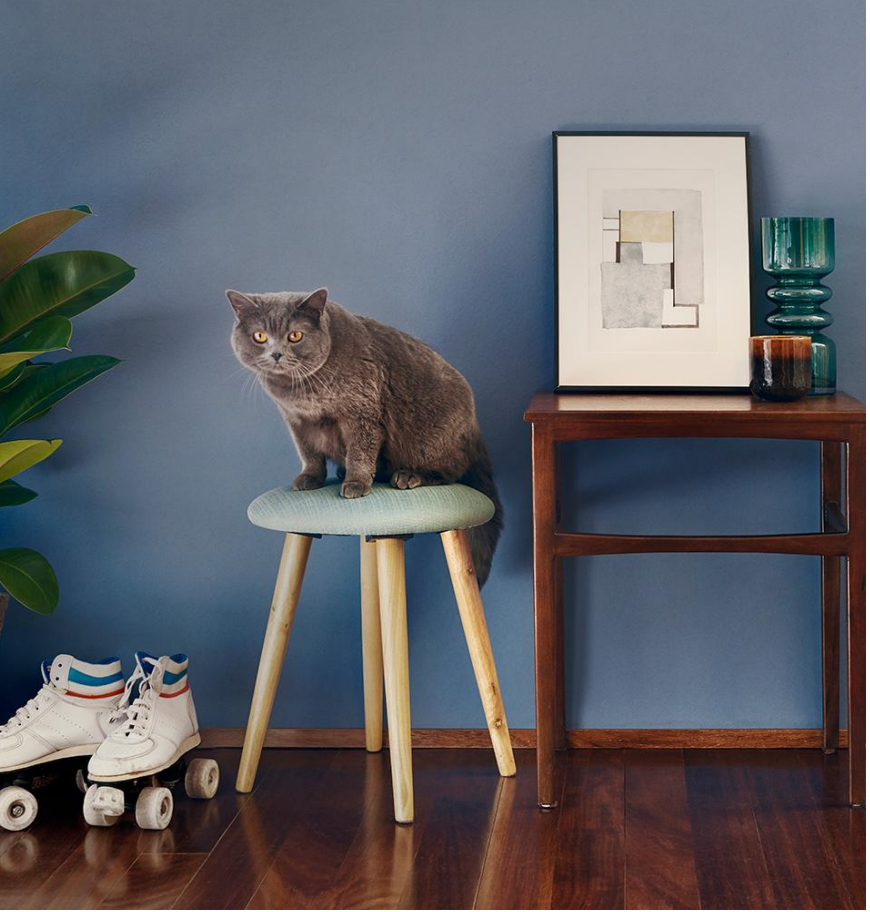

tibber

Hi <name>!
Thanks for accepting this Code Challenge.

We are Tibber

... the first digital energy company, enabling consumers to benefit from new technology.





Why a code challenge?

We are specifically looking to see your coding skills through your solution, as well as the process you use to get there.

This task will help us assess a number of criterias:

- Coding skills, both front- and back-end
- UX implementation
- Code style

The brief

Task

Create a web page using Vue.js or React.js that displays a graph of today's weather (temperatures) and couple of text and image blocks.

The web page should be backed by a simple web server based on Node.js. (NextJS, Koa, Express...)

The back-end should query today’s weather from our app-gateway GraphQL server (App API) and provide an interface for the front-end to consume. You can assume that the API is rate-limited, so you will need to cache the data and auth token.

Select the right tools for the job, you’re free to use whatever libraries or frameworks you find suitable for the task.

This Figma design is what we are visually aiming for, and the web page should be usable on both mobile and desktop.

The devil is in the details, and we love to see your own touch of UX, feel free to add some valuable user interactions. On that topic, we've added a magic link, with one purpose. This button should trigger something on the server-side that results in something visible in the front-end. Be creative: data-transformations, use other parts of our api, merging with data from a third-party api or just something crazy.

Please include a README file with instructions on how to start the solution, it’s also nice if you add some simple documentation and notes.

Fetch data

The data is accessible from a GraphQL-API. You can test it here:
https://app.tibber.com/v4/gql

Username/password:
demo@tibber.com/Electric

Once you are logged in, you can paste the query to the right in the editor. It will get today's weather for one of the demo user's homes.

```
{
  me {
    home(id: "a8c210fc-2988-4f06-9fe9-ab1bad9529d5") {
      weather {
        minTemperature maxTemperature entries {
          time temperature type
        }
      }
    }
  }
}
```

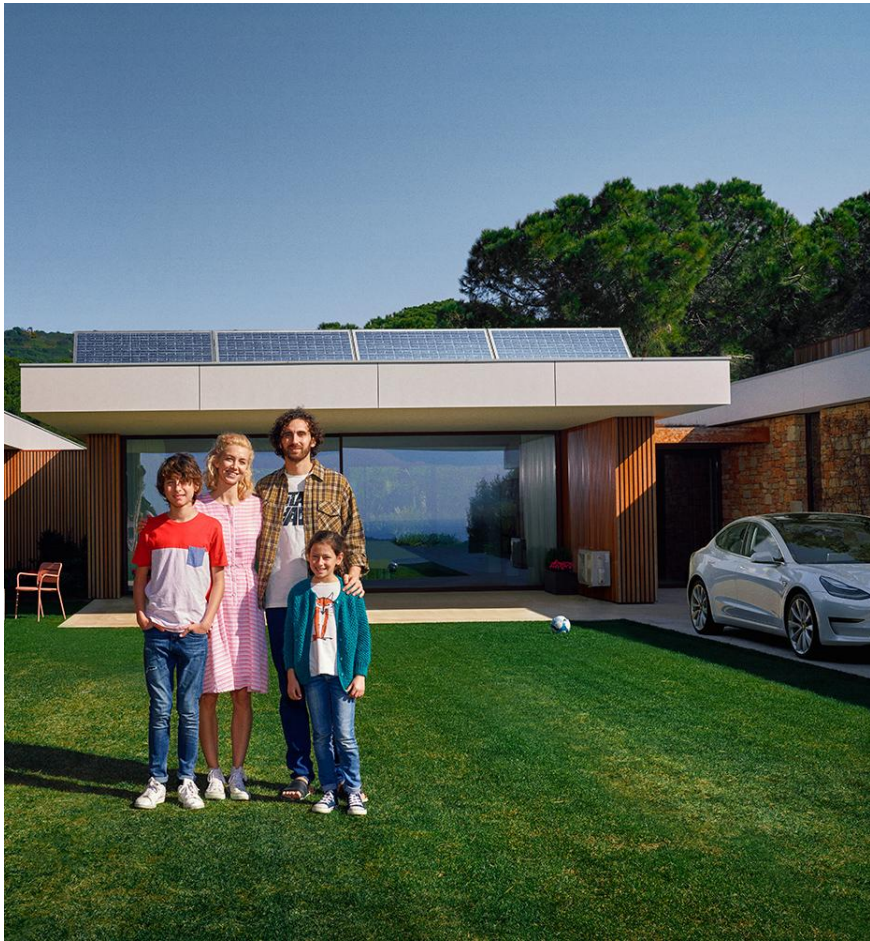
The raw representation of the weather-query above:

```
POST /v4/gql HTTP/1.1
Host: app.tibber.com
Content-Type: application/json
Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImI0MjAwMDFkLTE4OWItNDRjMC1hM2Q1LW
Q2MjQ1MmJmZGQ0MiIsImlzSW1wZXJzb25hdGVkIjp0cnVlLCJpYXQiOjE1ODQ3MDY4NjV9.S-4SVcZI g_MXcXfd8slLpAfJ8E565wMRFEqO4cDP3wc
{
"query":
"{me{home(id:\"a8c210fc-2988-4f06-9fe9-ab1bad9529d5\"){weather{minTemperature,maxTemperature,entries{time,temperature,type}}}}}"
}
```

Nice to know

In the raw representation of the weather-query there is an authorization-header with a jwt-token (JSON Web Token).
The jwt-token must be included to be able to access the api. To retrieve a token you can issue the following query:

```
POST /v4/login.credentials HTTP/1.1 Host: app.tibber.com
Content-Type: application/json
{
        "email":"demo@tibber.com", "password":"Electric"
}
```



Thanks!

For questions please reach out to tomek@tibber.com