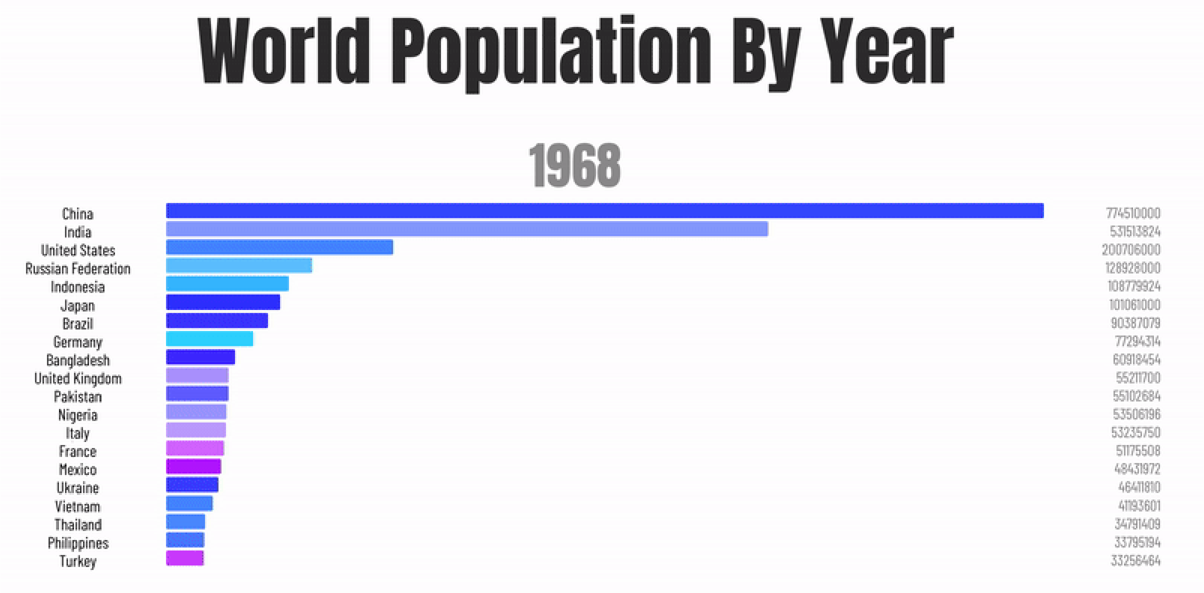
Frontend - carbon footprint chart task

👋 Welcome to the Altruistiq front-end hiring task! We're very much looking forward to see how you tackle and code this task. It should take you about 2 hours, but you're free to spend as much time as you like. Once you submit this task and if we think it looks well executed, we will discuss it together and go through a couple of related technical concepts.

Objective

Create a chart animating carbon footprint per country throughout the years. The below chart is an example for world population. The provided Figma file shows the one-page app that you should build for this task.



Requirements

1. pixel perfect implement the provided Figma file as a Vue app
2. use data from the Footprint network and store in Vuex
3. cache data with timeout of 5 minutes, making sure upon reload the app conditionally uses stored data
4. use either components, native DOM or D3 to create the chart.
5. animate the sorting and bar length of the chart as per the provided example.

6. chart should be responsive, test by changing the window size (dragging)
7. calculate and show world total footprint on the page
8. use Sass/Scss
9. do not use chart / css / component / animation / caching libraries. The idea is that you show your skills by coding this manually.

Get started

1. Get this Figma file. The ... in the chart is a placeholder. The idea is that you show all the countries from the API response in the chart.

2. Request an API key from https://data.footprintnetwork.org/#/api, you will receive it in 5 minutes by email

3. API authentication uses Basic Auth, example with Axios:

```
await axios.get(apiUrl, {}, { auth: { username: 'any-user-name', password: API_KEY } })
```

3. Use the endpoint `GET https://api.footprintnetwork.org/v1/countries` to get the list of countries and `countryCodes`, example response:

```
[ { "id": 1, "version": null, "countryCode": "1", "countryName": "Armenia", "shortName": "Armenia", "isoa2": "AM", "score": "3A" }, { "id": 2, "version": null, "countryCode": "2", "countryName": "Afghanistan", "shortName": "Afghanistan", "isoa2": "AF", "score": "3A" }, ... ]
```

What you need is the `countryCode` and `countryName`

4. Use the `countryCode` from step 3 to query the endpoint `GET https://api.footprintnetwork.org/v1/data/{{countryCode}}/all/EFCpc` This will return all available data for a specific country, example response:

   ```
   [ { "id": 4193, "version": null, "year": 1961, "countryCode": 10
   , "countryName": "Australia", "shortName": "Australia", "isoa2":
   "AU", "record": "EFConsPerCap", "cropLand": 0.521280004829556,
   "grazingLand": 2.81094109134558, "forestLand": 1.03202880039184,
   "fishingGround": 0.0509710310900806, "builtupLand": 0.0247025489
   319933, "carbon": 3.02800424434668, "value": 7.46792772093573,
   "score": "3A" }, ... ]
   ```

   Do this for all countries. To chart the carbon footprint for a specific year, you need the `carbon` attribute from the response.

5. Cache results in browser with a timeout of 5 minutes. This means that if someone refreshes the app within the 5 minutes timeout, the app loads data from the cache. Once the cache expires, it queries the endpoints again.

6. Build the page as per the Figma design, and make the chart functional as per the example GIF

7. Used fonts

   ```
   @import url('https://fonts.googleapis.com/css2?family=Inter:wght
   @300;400;500;600&display=swap');
   ```

If you have any questions please do not hesitate to reach out. Good luck!