

S2I FRONT-END FINAL PROJECT
MADE BY ANTHONY DEI MEDICI



iEarth

Learn about climate change main factors

TRY ME

Github repo: <https://github.com/AnthonyDM-Dev/global-warming-dashboard>

iEarth

Start

Summary

- Mission [Pag. 03](#)
- App Description
 - Cover [Pag. 05](#)
 - Introduction [Pag. 06](#)
 - Dashboard [Pag. 07](#)
 - Child Page [Pag. 08](#)
 - Chart Filters [Pag. 09](#)
 - Searching Form [Pag. 10](#)
 - Error Handling [Pag. 11](#)
- Technical Info
 - Framework [Pag. 13](#)
 - Mixins [Pag. 14](#)
 - NuxtChild [Pag. 15](#)
 - Deploy [Pag. 16](#)
- Contacts [Pag. 17](#)
- Thank you [Pag. 18](#)

Mission

Realize a multipage dashboard app in order to
visualize data about global warming



App Description

Cover

This is where the user lands for the first time.

This cover page let me load part of the document.

A spinner is present until the iFrame script containing the 3d model has been downloaded.

Then, a start button appear.

iEarth



iEarth

Start

Introduction

Next, a small intro appear.

For this introduction, I animated some texts in order to make them appear in cascade.

Background sounds start. The user can silence sounds by pressing the audio button.

A "skip" button is present, in case the user has already seen the intro.



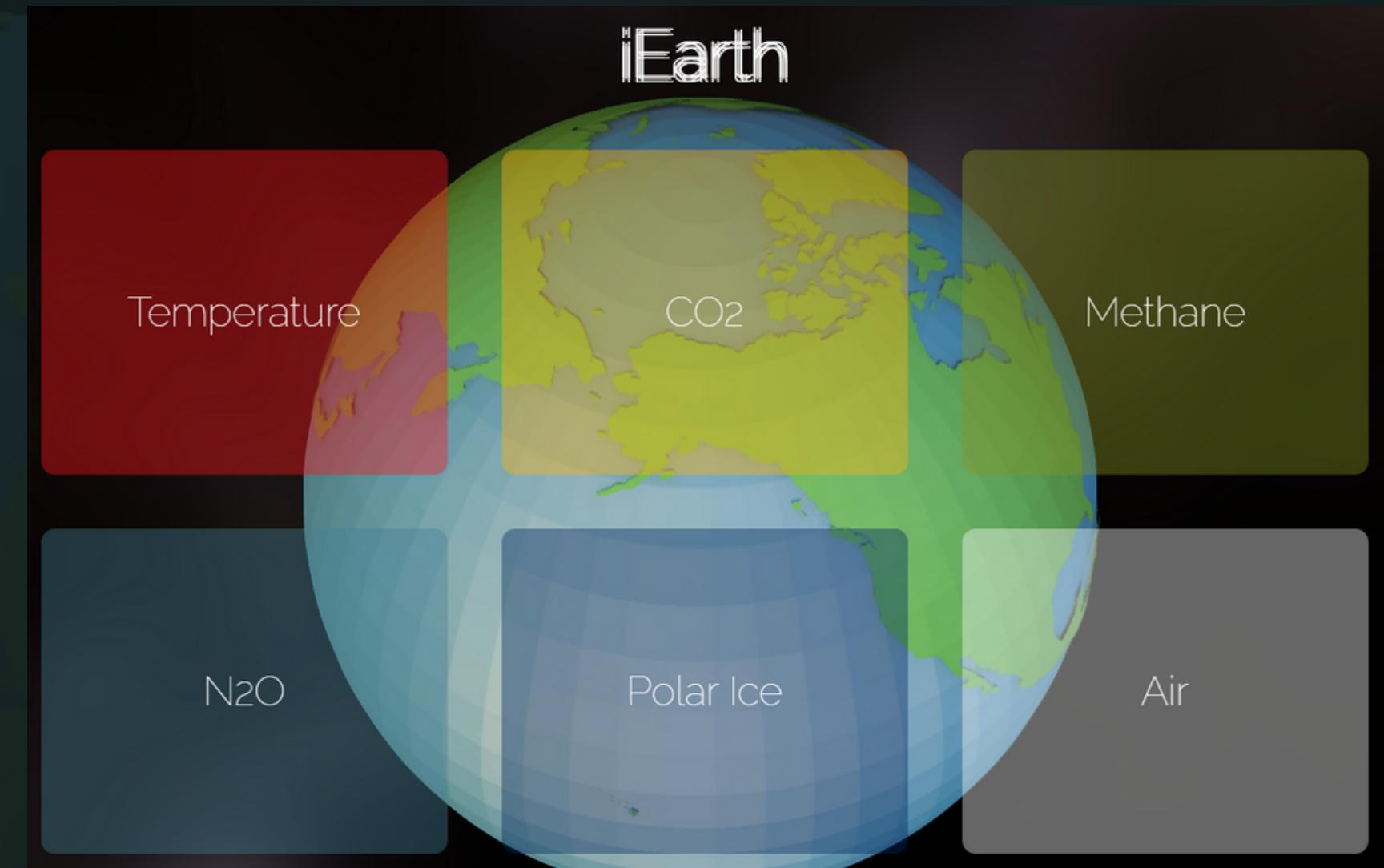
Dashboard

Here, the user can learn about global warming by diving into each voice

5 five cards appears fading-in.

Each cards has text content on its back. User can read it by clicking (mobile) or hovering (desktop) on it and the card will flip.

A "dive in" button is present to know more about the single topic.



Child Page

By clicking on each topic a bigger window appear.

The page is composed by:

- A heading with a video autoplaying in the background
- Short description
- Chart reporting some stats about the topic selected. Made with ChartJS.
- In some pages, a form handling API responses can be found.
- A Youtube video embedded in the page.
- Back button to return on the dashboard.
- Side arrows to navigate in each topic without returning to the dashboard each time.

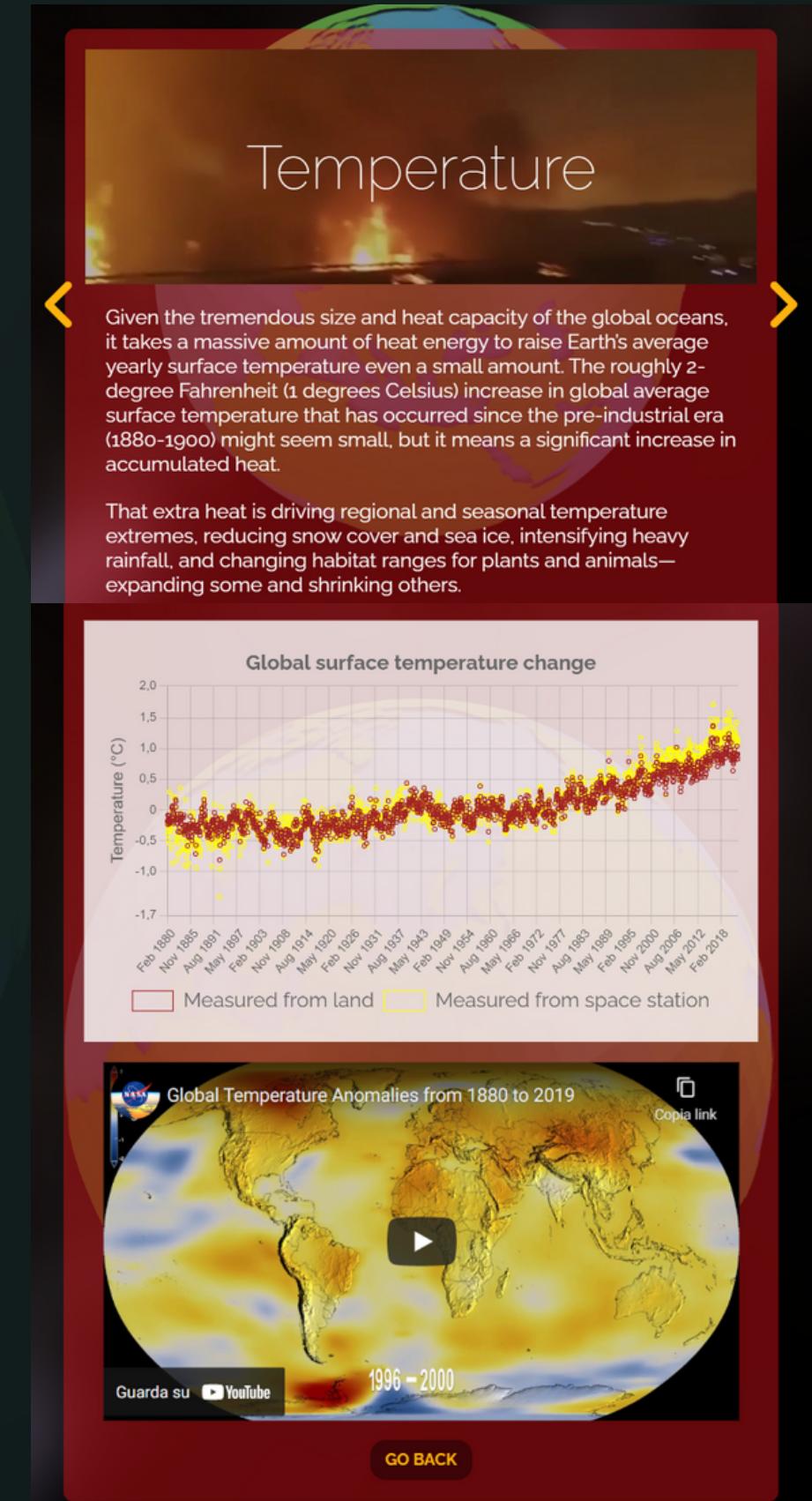
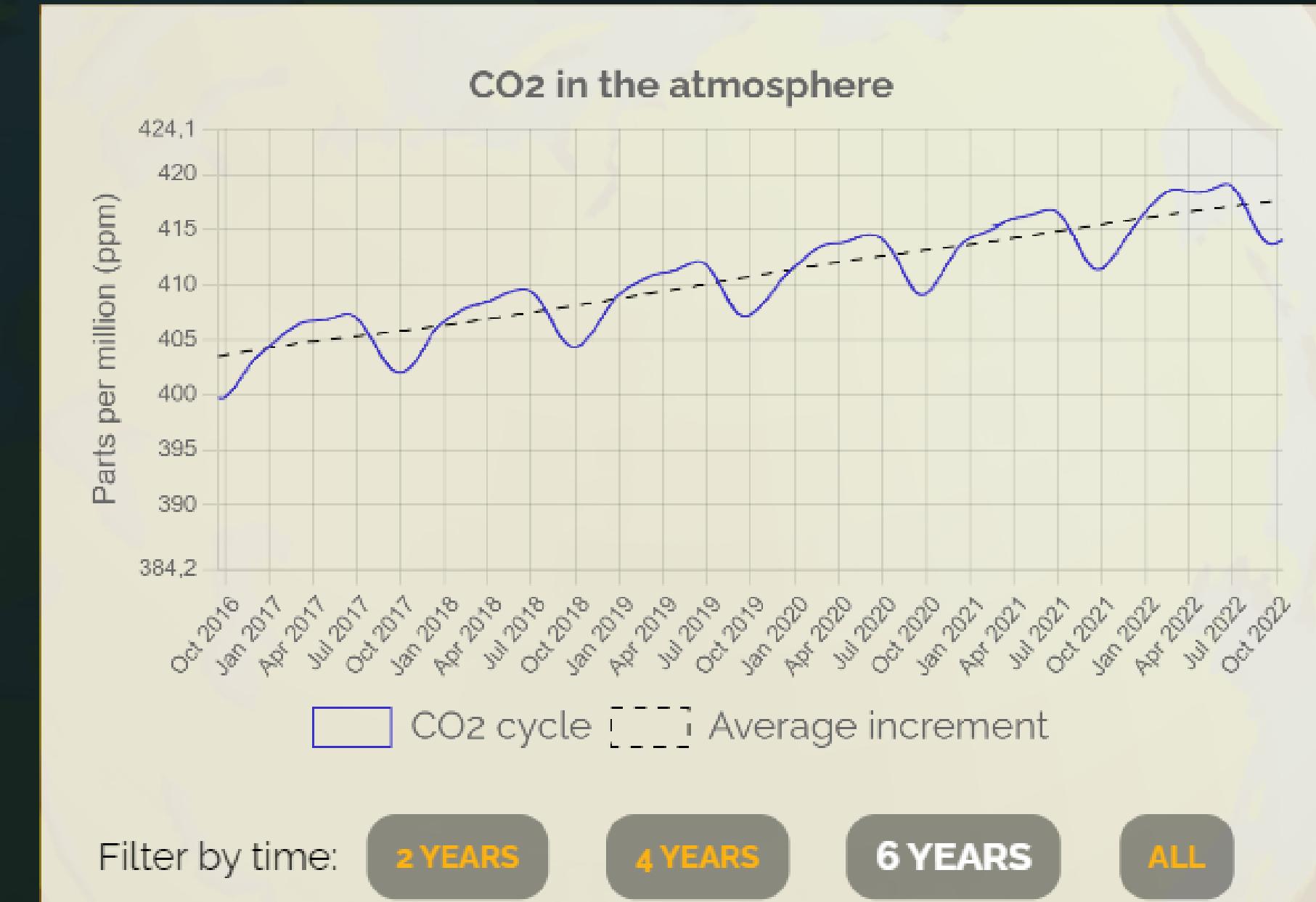


Chart Filters

Time filters has been provided to every chart.

Multiple time filters let the user expand the informations inside every chart.



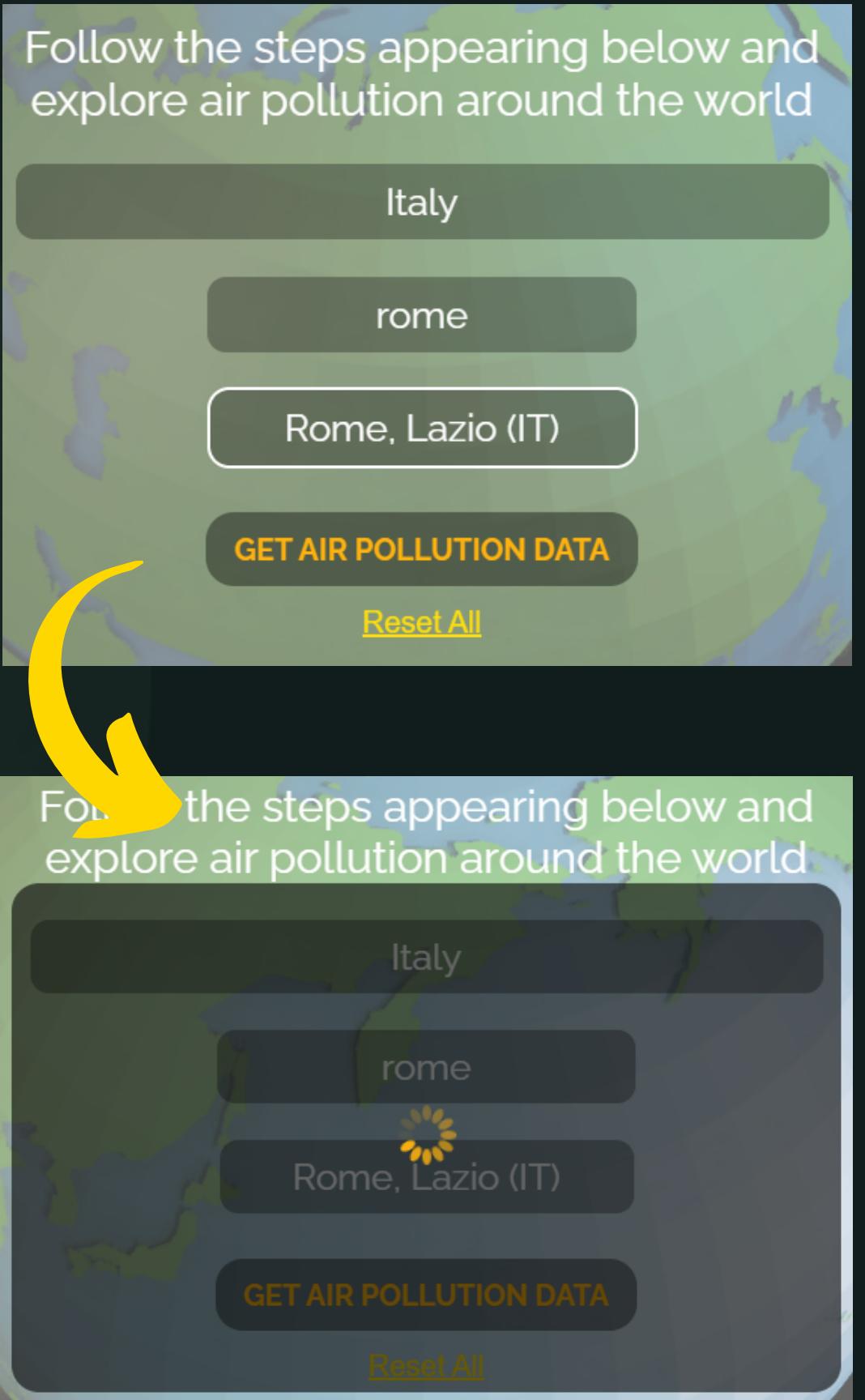
Searching Form

In the Air page, a form can handle city searchings by user text inputs.

This form handles two API responses in cascade:

- OpenWeather Geocoding, helps the user to be more specific about the city searched and get its latitude and longitude.
- OpenWeather History, get the air quality history about the city searched based on its coordinates.

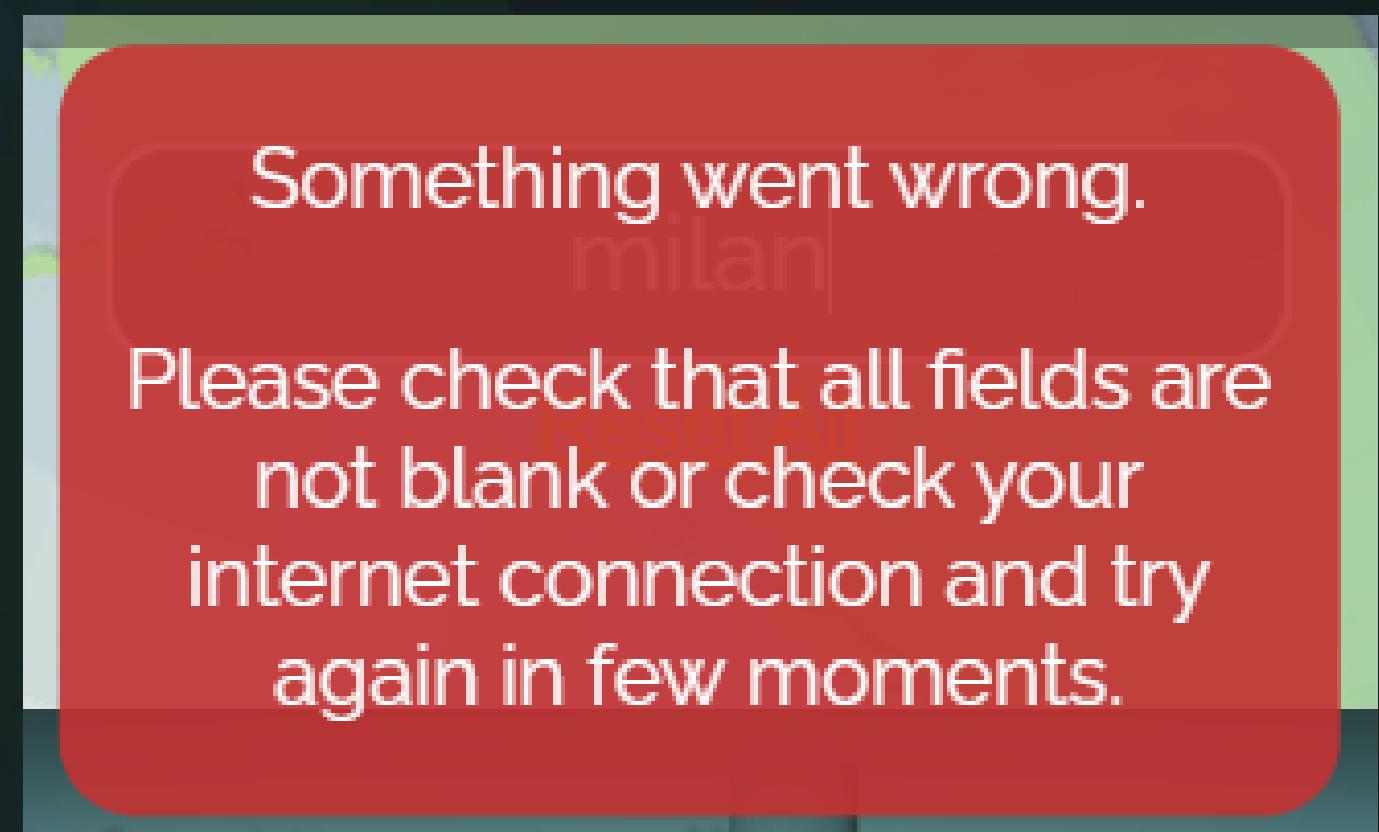
A spinner component has been used to inform the user about the loading state.



Error Handling

An error popup shows up in case of API errors.

Every time an API does not retrieve the data needed to load the chart, a popup error shows up.





Technical
Info



NUXT

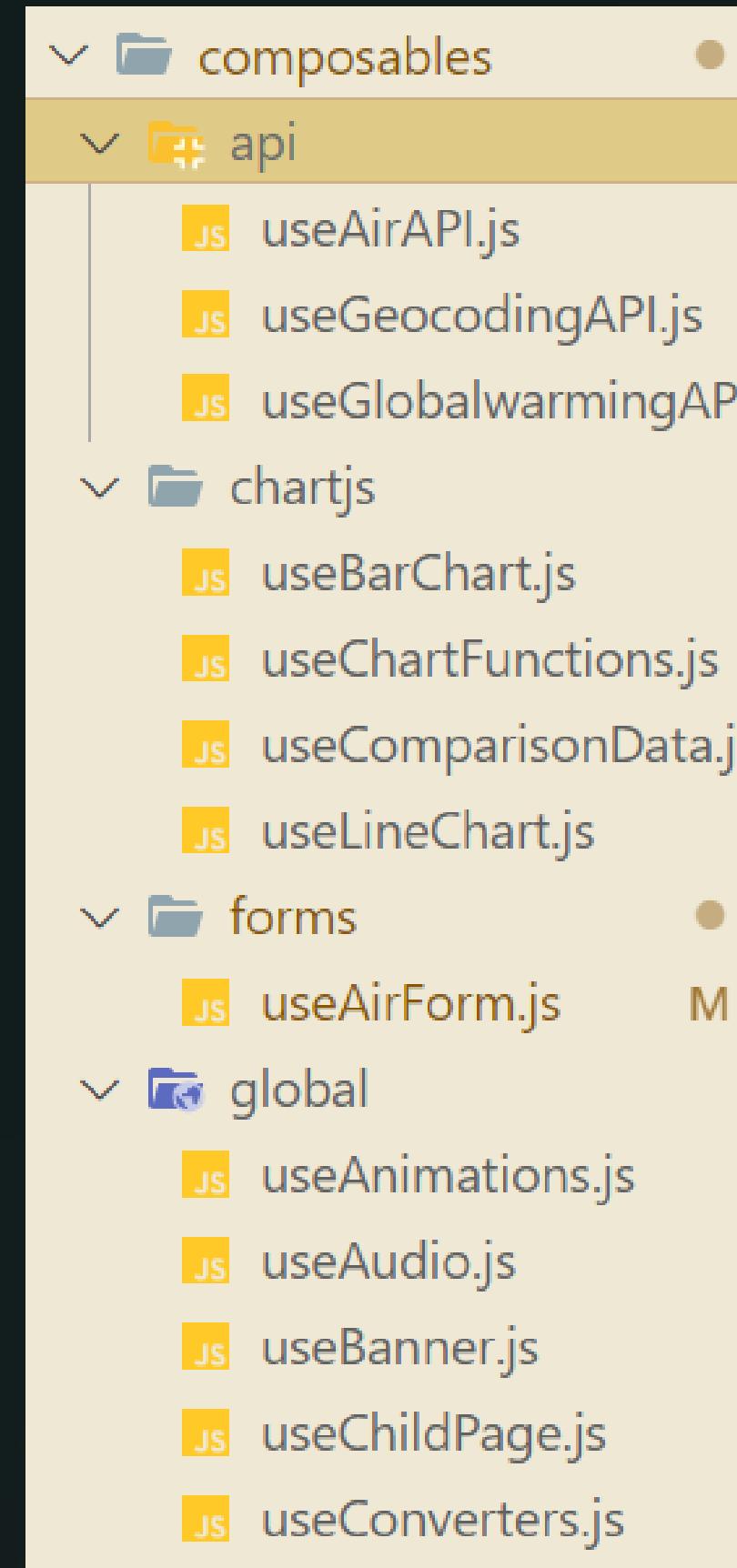
Framework

The framework I decided to use is NuxtJS.

It offers a lot of quick features to design code fast and clean thanks to the Composition API Nuxt library.

Routing is the first aspect of why I choose it.

In order to create a versatile multipage app I used the NuxtChild feature.



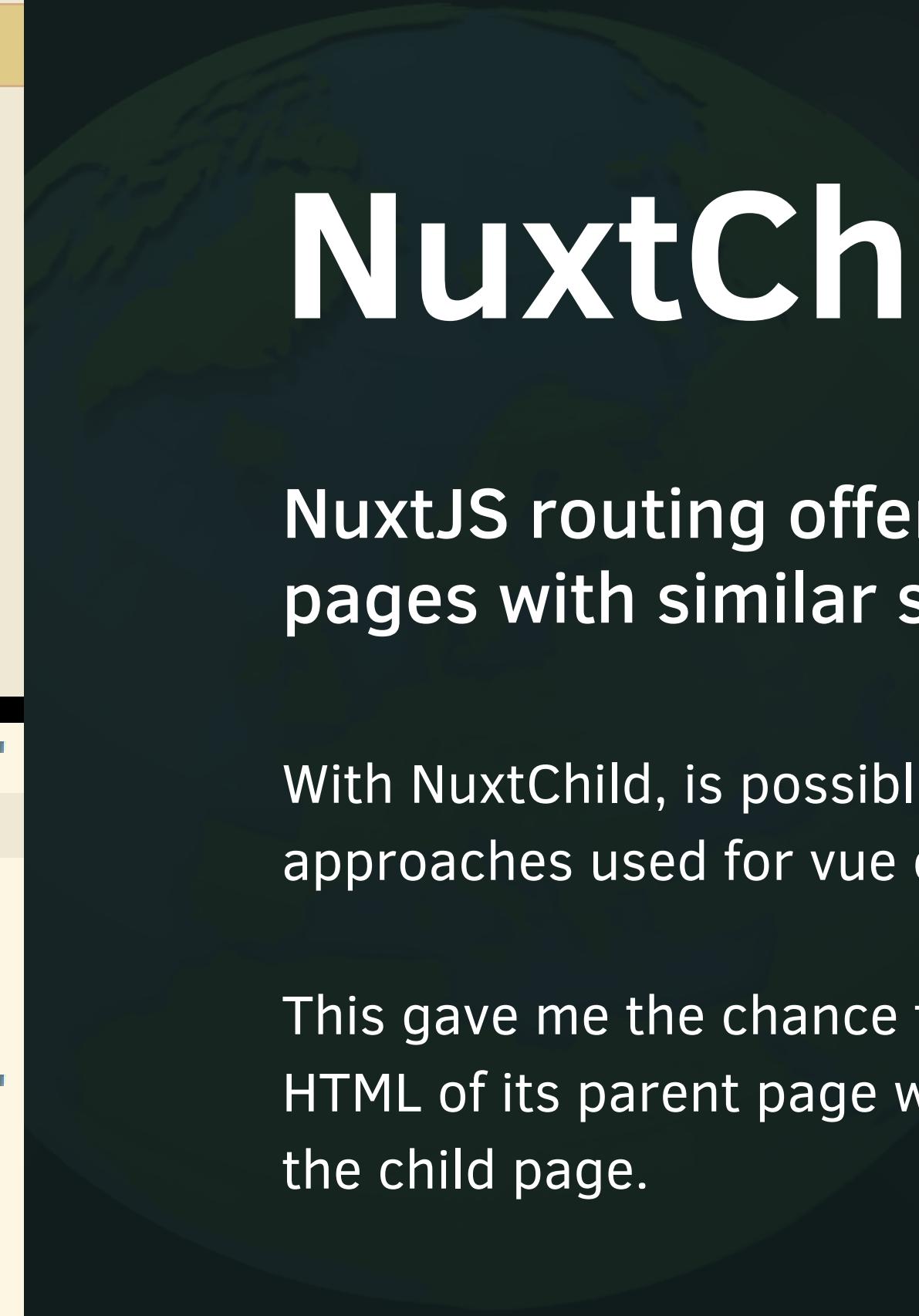
Composition API

VueJS 3 provide a powerful tool to modularize your code.

Thanks to Composition API I avoided redundance code in every NuxtChild page.

Compared to Options API, Composition API offers better code organisation and a better implementation of logics.

In case of bigger and complex projects, Composition API is preferable to Options API.



```
<div v-if="childPage.isVisible">
  <NuxtChild
    class="fade-in"
    :title="pageTitle"
    :video="pageVideo"
    :youtube-link="pageYoutube"
    :color="pageColor"
    @go-to="returnToHomepage"
  />
```

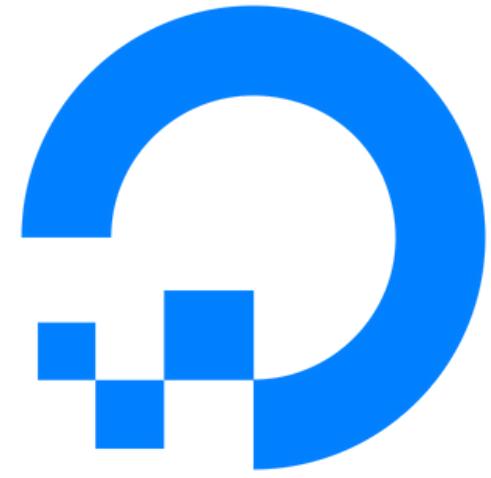
A file tree and code snippet illustrating NuxtChild usage. The file tree shows a 'pages' directory containing an 'index' folder with several Vue files: CarbondioxidePage.vue, MethanePage.vue, NitrousoxidePage.vue, PolaricePage.vue, TemperaturePage.vue, and index.vue. Below the file tree is a code block showing the use of the NuxtChild component in a parent page's template.

NuxtChild

NuxtJS routing offers a nice way to handle pages with similar structure.

With NuxtChild, is possible to apply to vue pages similar approaches used for vue components.

This gave me the chance to create new pages reusing the HTML of its parent page without the need of duplicate it in the child page.



DigitalOcean

Deploy

I used DigitalOcean droplets to deploy my dynamic site.

I usually used Netlify or Vercel to deploy my static apps but I have had to use DigitalOcean droplets in this case in order to easily handle this Single Page Application app.

Contacts

Click on the icons below and
lets keep in touch!





Thank you for
watching