

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](#)
- Technical Info
 - Framework [Pag. 10](#)
 - Mixins [Pag. 11](#)
 - NuxtChild [Pag. 12](#)
 - Deploy [Pag. 13](#)
- Contacts [Pag. 14](#)
- Thank you [Pag. 15](#)

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 below without the user noticing.

A start button appear only when the iFrame script containing the 3d model has been loaded.

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.



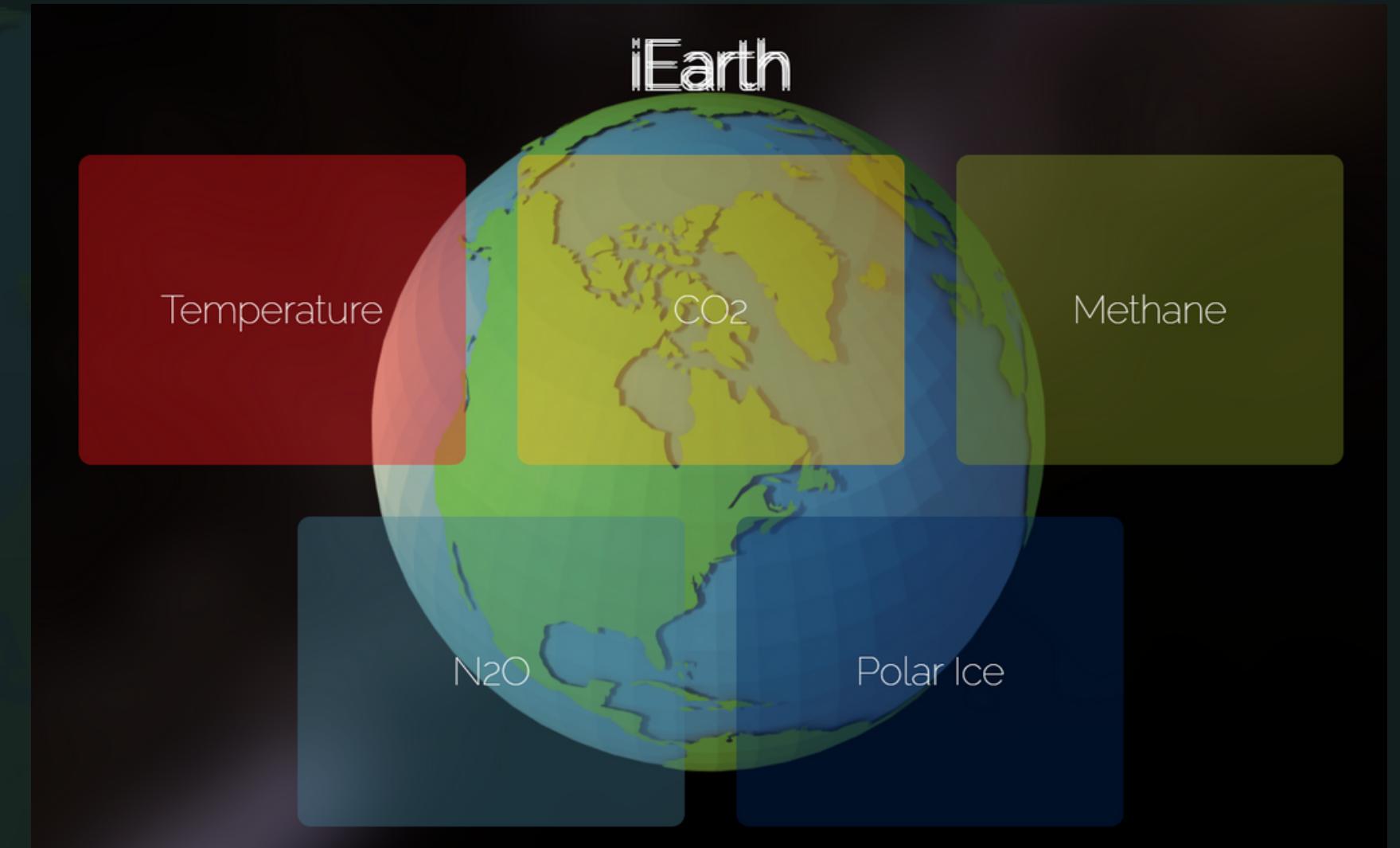
Dashboard

Here, the user can learn about global warming by dive 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.

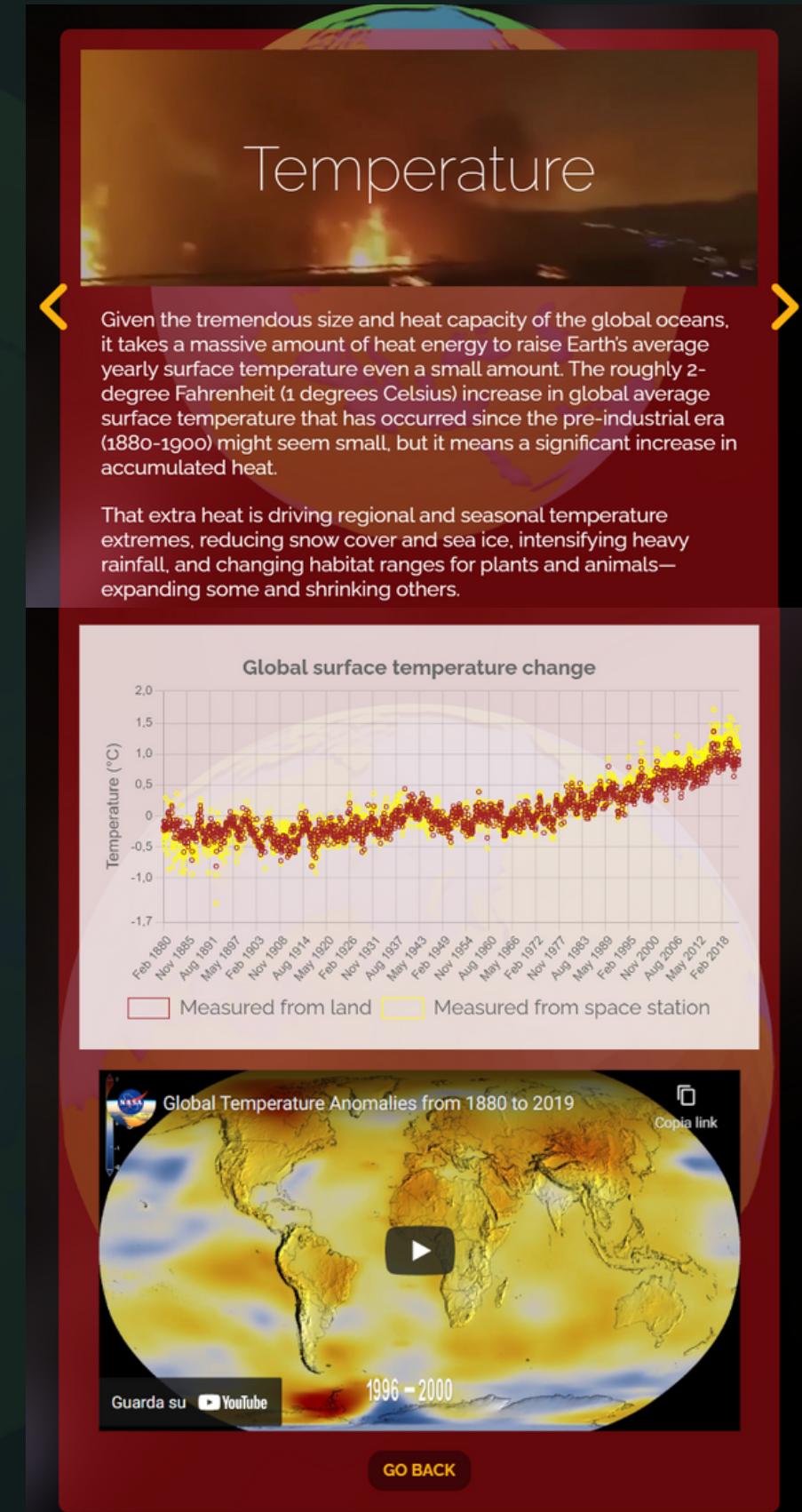


Child Page

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.
- 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.





Technical
Info



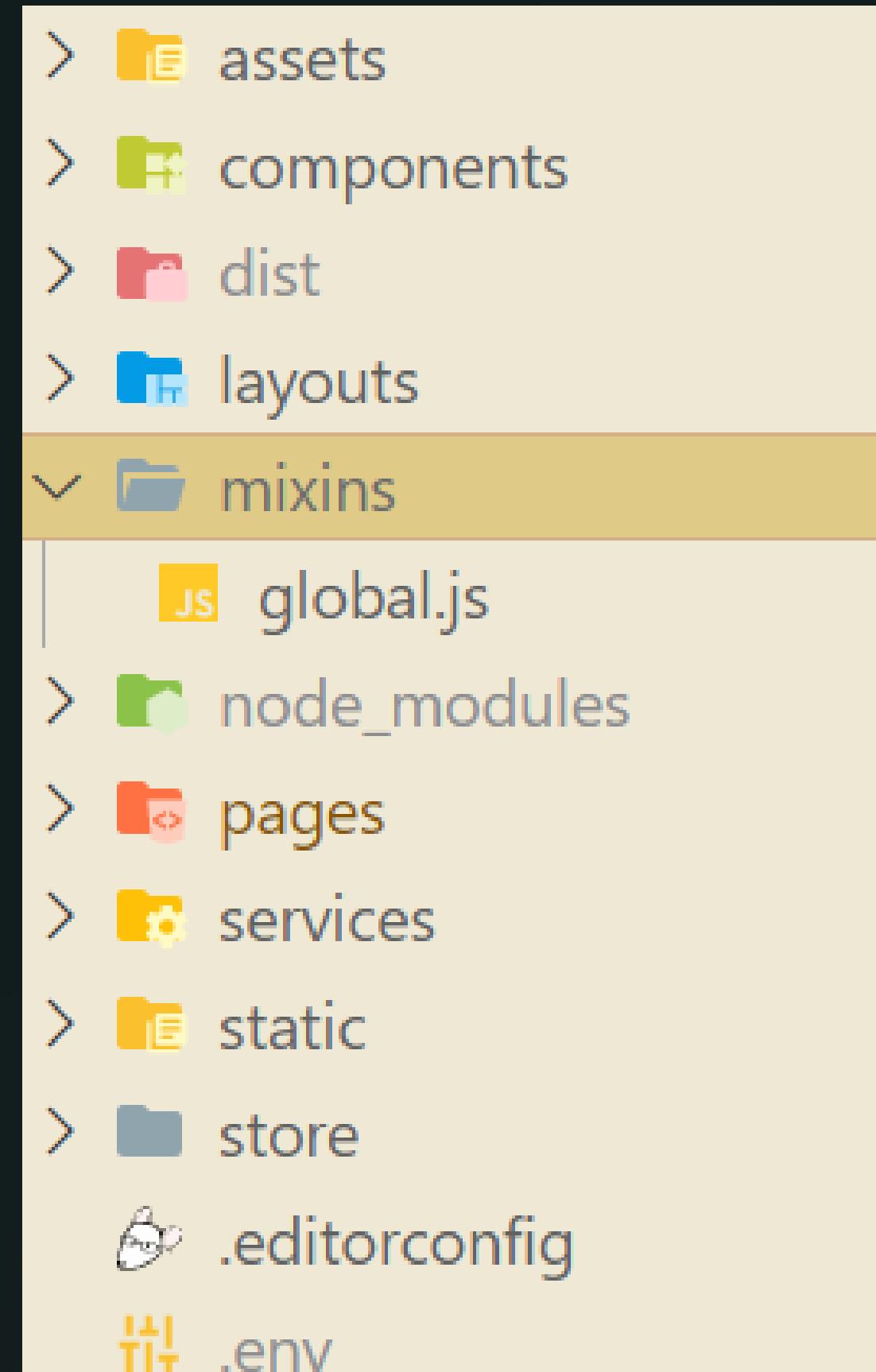
Framework

The framework I decided to use is NuxtJS.

It offers a lot of quick features to design code fast and clean.

Routing is the first aspect of why I choose it.

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

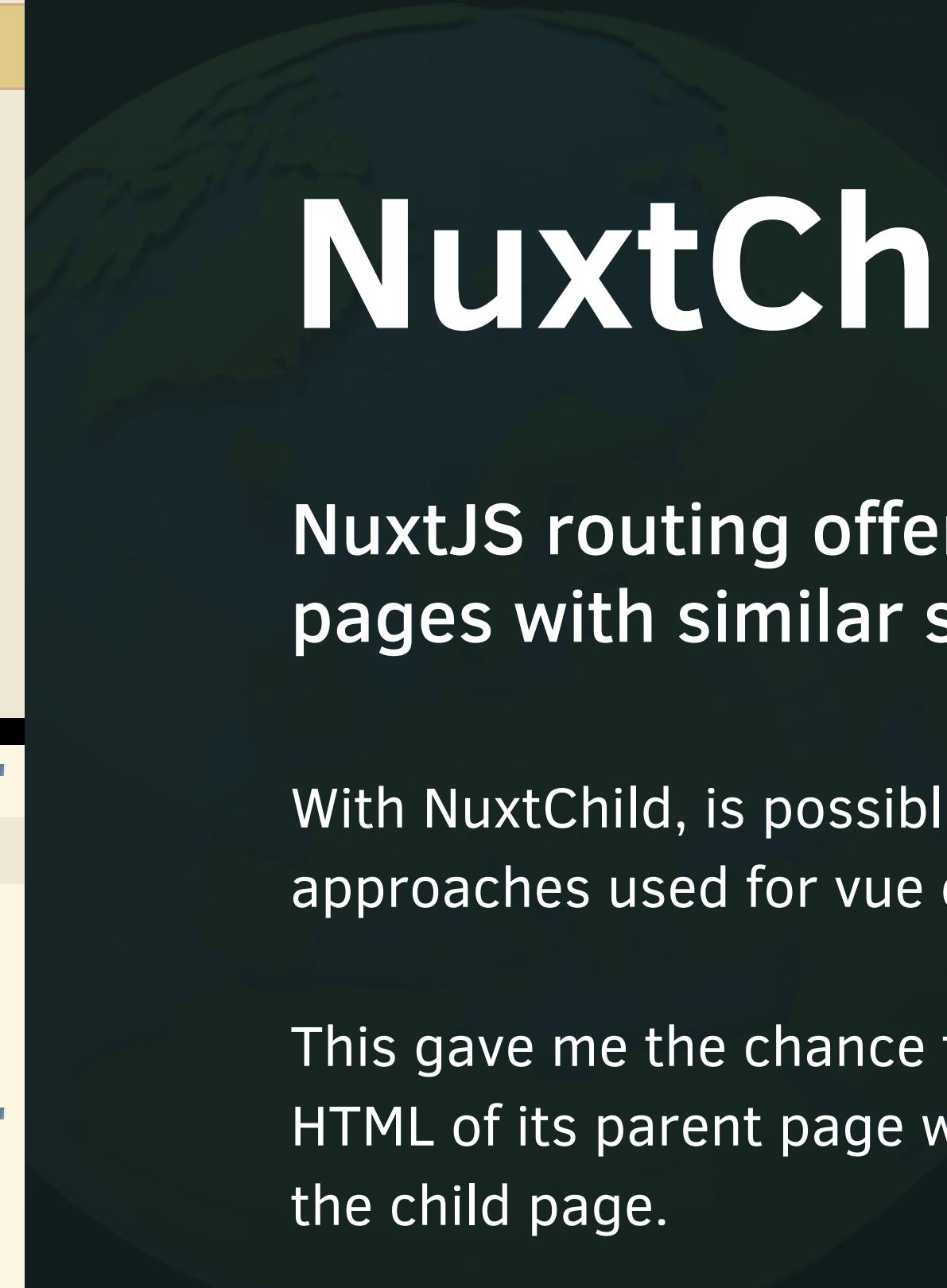


Mixins

VueJS provide a simple way to modularize JS code

Thanks to mixins I avoided redundancy code of every NuxtChild page.

This makes it also easier to read.



```
<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.



Deploy

I used Vercel to deploy my dynamic site.

I usually used Netlify to deploy my static apps but I have had to use Vercel 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

