Next.js & Netlify-cms

Blog site build

David Richards - February 2021

jamstack?

- Javascript, API (application program interface), Markup (html, css)
- Uses “someone else’s” compute resources
  - No need to run backend server (Netlify, Vercel, AWS, Azure, GCP, DigitalOcean, Heroku)
  - Configure for automatic scaling (up or down demand)

Reactjs

- Reactjs is often referred to as a component library
  - Since it has a set of rules it probably should be considered a framework
  - React apps are built from code components (button, navbar, page)
  - React is very un-opinionated for constructing projects allowing use of many different methods & libraries (e.g. routing: react-router, reach-router)
  - Flexibility can result in complexity of choices
  - Functional components replacing class components
  - React Hooks(functions) connect to state and lifecycle features
  - Permits different styling methods
    - CSS-modules, JSX, styled-components, global styling, css-in-js

Nextjs

- Framework around React
  - Static Site Generation
    - Creates complete code bundle delivered to browser
      - No interaction except navigation
  - Server Side Rendered
    - Builds in response to browser request - complete page
    - Client Side rendering delivers initial HTML then loads javascript (hydrates)
    - Hybrid
      - Static (product page), API (Price data)

Nextjs Benefits

- SEO - Rendered pages actually contain all html
- Code splitting
    - Only loads JS & CSS for viewed page
- HMR - hot module replacement - only reloads changed files
- Built-in routing - just add page
- Built-in CSS support
- API routes for API endpoints with Serverless Functions

SEO Demo

- Open VSCode
- Launch terminal (ctrl + `)
- Navigate to your projects folder/directory
- >npx create-react-app react_example
    - >cd react_example
    - >npm run dev
    - Will open web page at localhost:3000
    - Inspect page with Dev Tools
        - Will show an empty div - no content for Google Bots

SEO Demo

- Navigate to your projects folder/directory
  - >npx create-next-app next_example
  - >cd next_example
  - Edit package.json
    - “dev”: “next dev -p 3001”
  - In new window open terminal (ctrl + `)
  - >npm run dev
  - Will open web page at localhost:3001
  - Right click on page and ‘Inspect’
  - Select Elements from menu to see the indexable HTML

Build out introductory project

Creating Project

- Three choices:
    - Create the app from scratch using create-next-app
    - Clone the main branch of repo and complete the project coding following instructions starting page 10
    - Clone the CMS-Integration branch for a complete built project

From Scratch - Create-Next-App

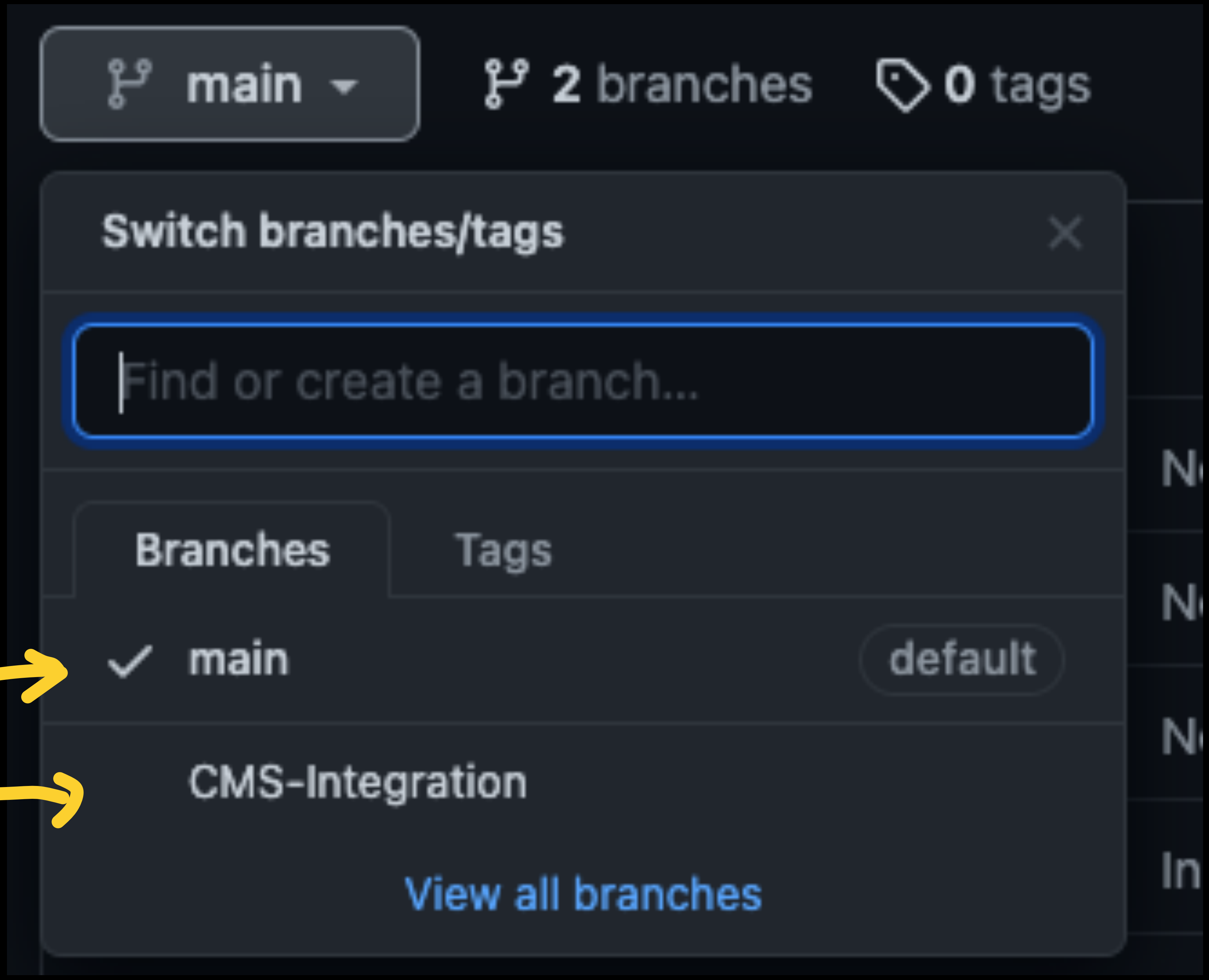
- npx create-next-app <yourProjectName>
- cd <yourProjectName> && npm run dev (make sure to kill previous examples)
- npm install - -save-dev frontmatter-markdown-loader
    - Alt: npm i -D frontmatter-markdown-loader
- npm install - -save-dev @babel/core @babel/preset-react
- mkdir content
- create content/horses.md && add content from example pages
- package.json
    - under scripts
        - “export”: “next build && next export”

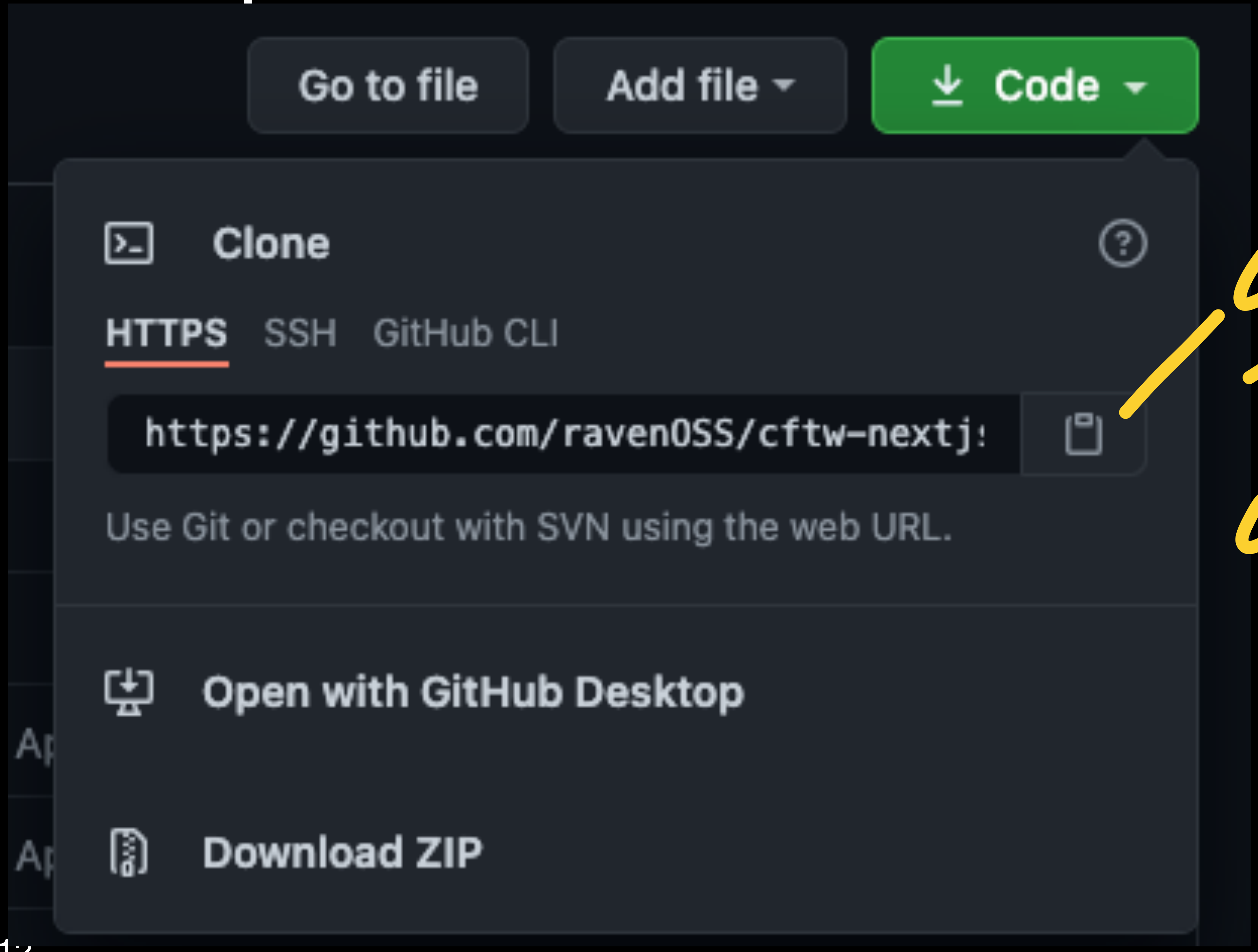
Deploy Repo Examples

- Clone repo
  - Main branch:
  - git clone https://github.com/ravenOSS/cftw-nextjs-netlifyCMS.git
  - CMS-Integration branch
  - git clone https://github.com/ravenOSS/cftw-nextjs-netlifyCMS/tree/CMS-Integration
  - cd cftw-nextjs-netlifyCMS
  - npm install
  - npm run dev
  - browser: localhost:3001

Clone Repo Guide

- There are two choices (see image below)
  - Clone the main repo and modify per the instructions in this PDF starting at page 10
  - Clone the CMS-Integration branch that will provide complete working source code. It will still require some setups to launch a live site.
-





```
https://github.com/ravenOSS/cftw-nextj:
```

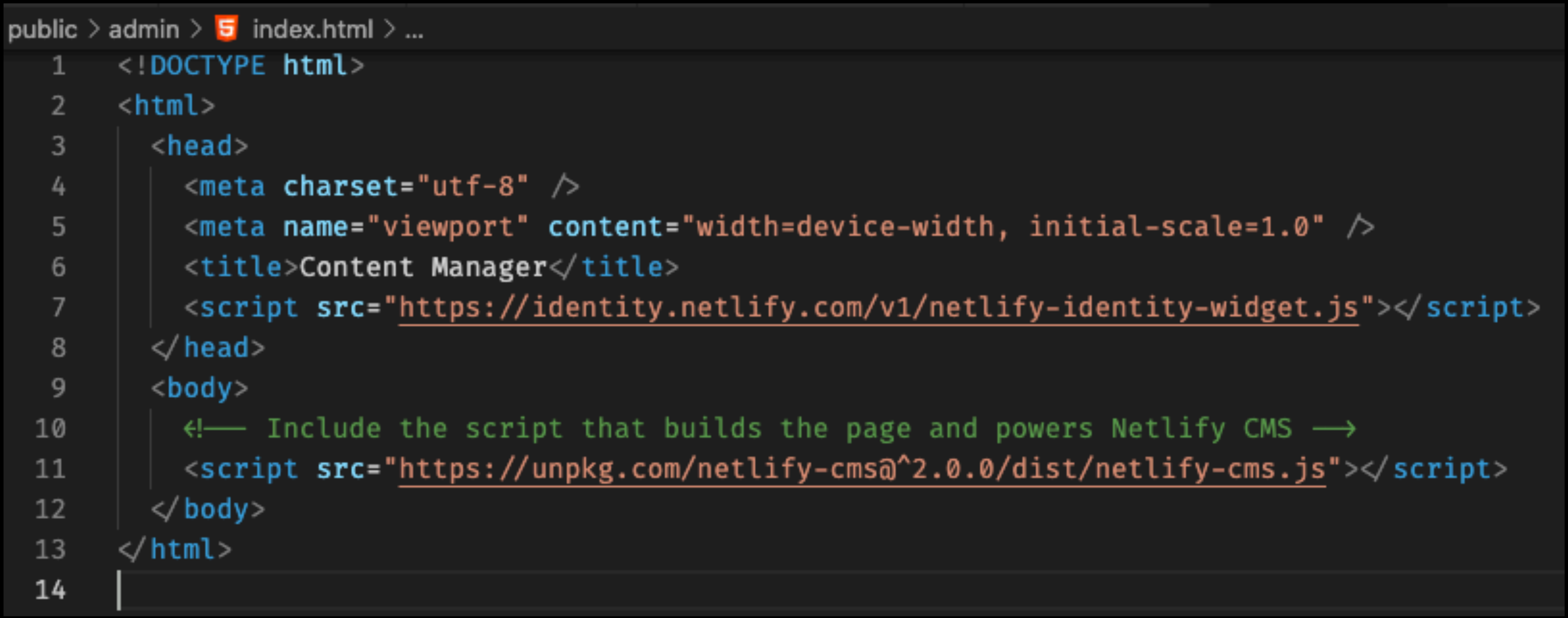
Use Git or checkout with SVN using the web URL.

Copy To Clip Board

Add netlify-cms

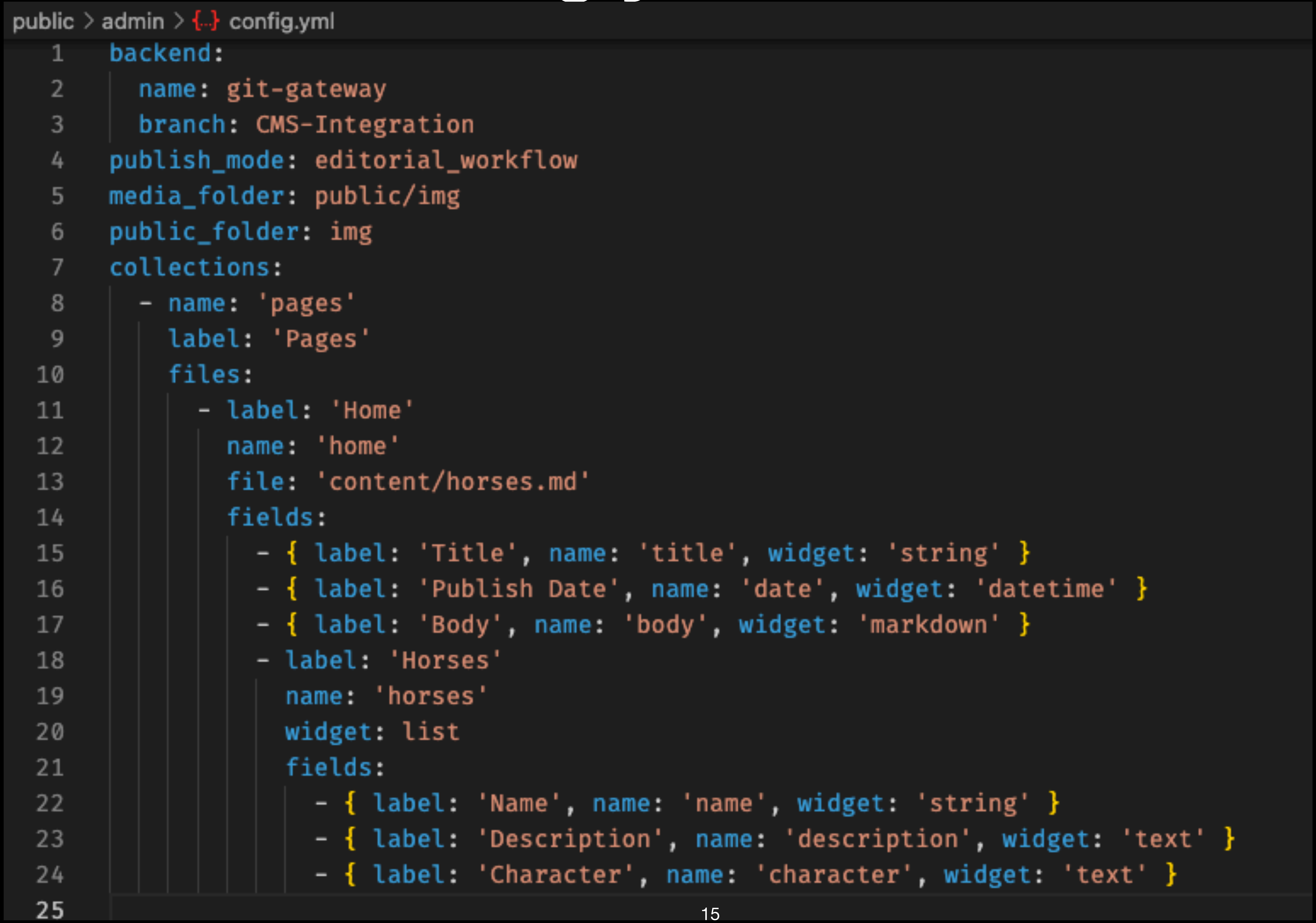
- create admin folder/(directory) in /public
- create two files
  - index.html
  - config.yml
- Add content from next two pages

public/admin/index.html



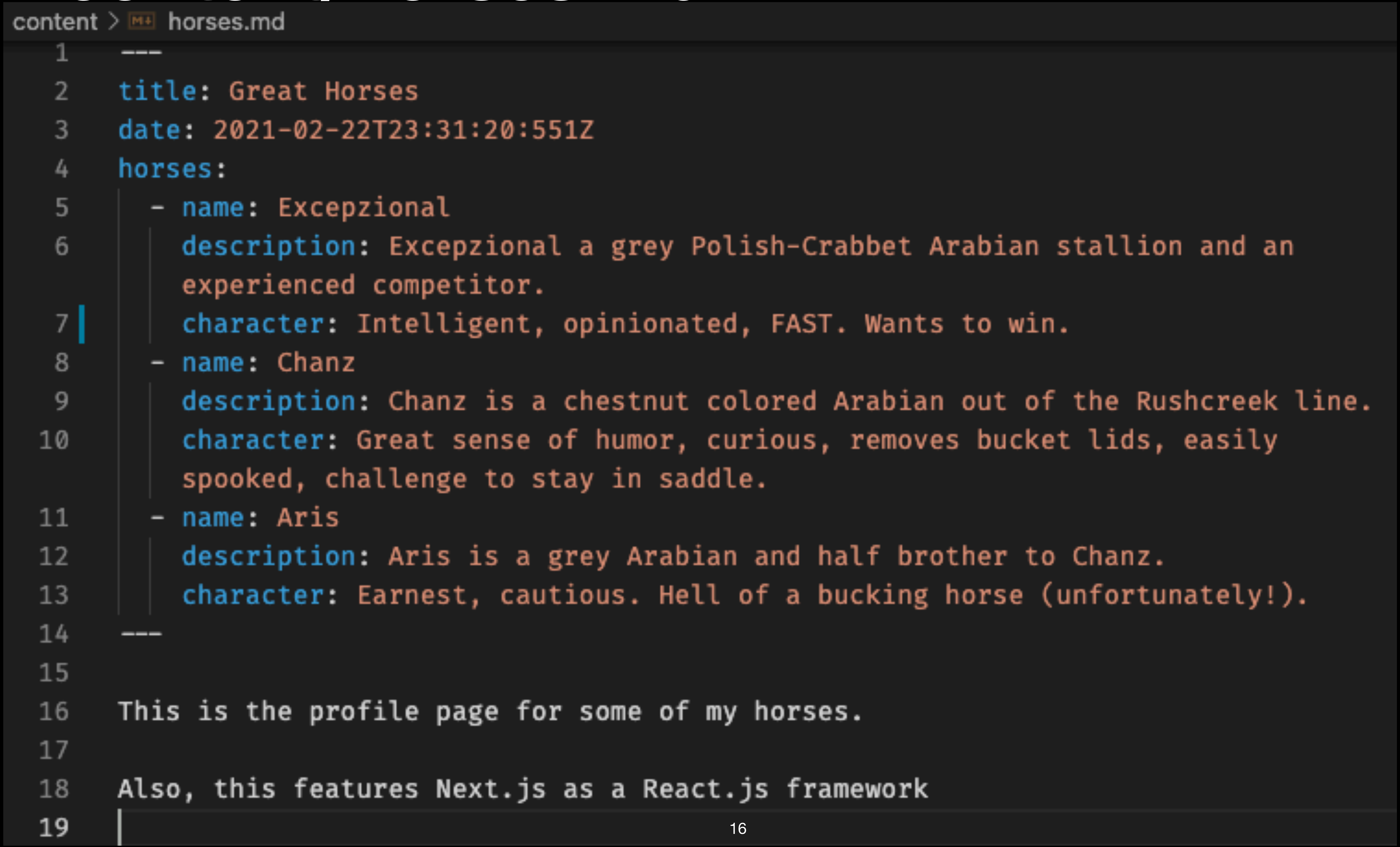
```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Content Manager</title>
    <script src="https://identity.netlify.com/v1/netlify-identity-widget.js"></script>
  </head>
  <body>
    <!-- Include the script that builds the page and powers Netlify CMS -->
    <script src="https://unpkg.com/netlify-cms@^2.0.0/dist/netlify-cms.js"></script>
  </body>
</html>

```

public/admin/config.yml



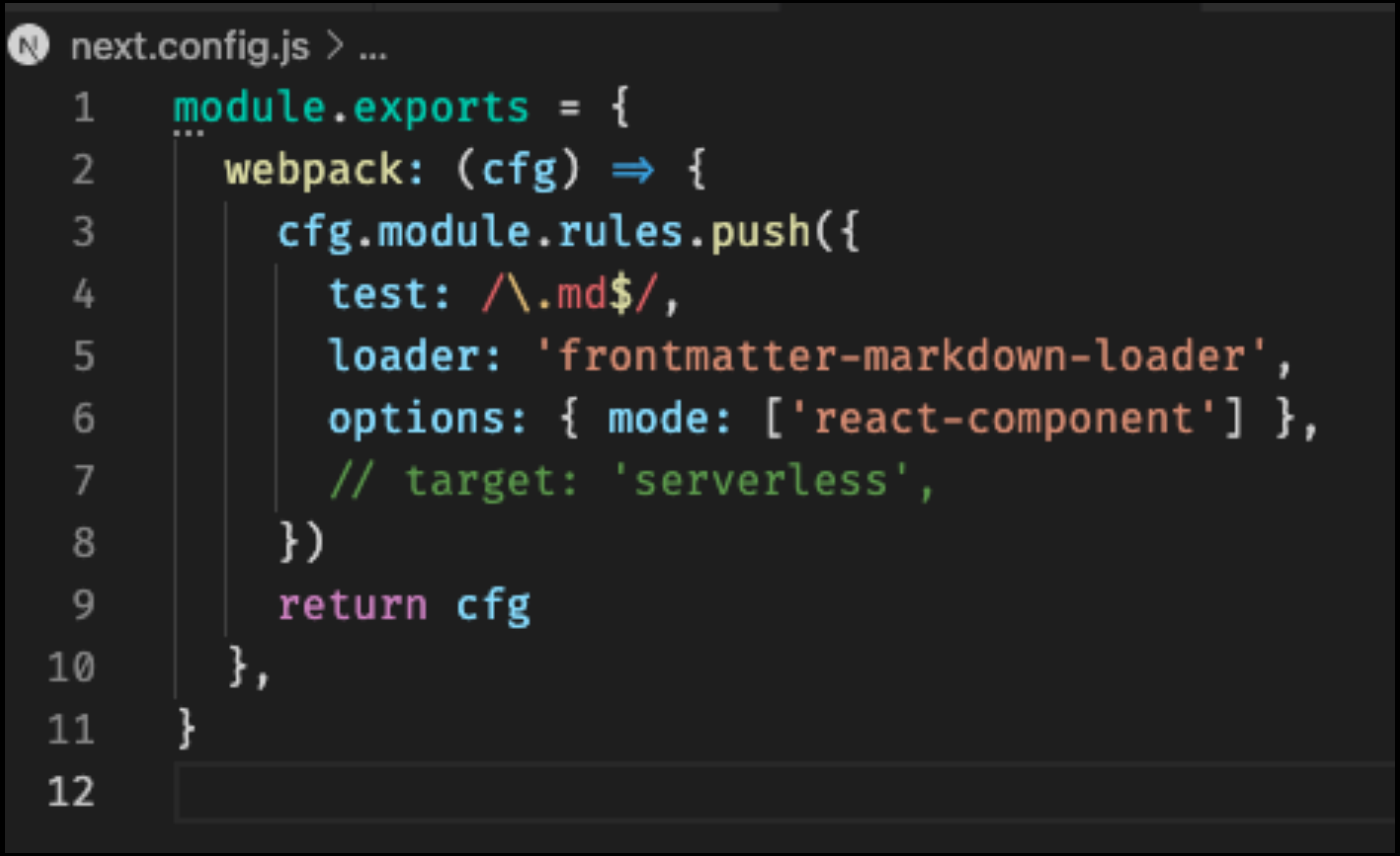
```yaml
backend:
  name: git-gateway
  branch: CMS-Integration
publish_mode: editorial_workflow
media_folder: public/img
public_folder: img
collections:
  - name: 'pages'
    label: 'Pages'
    files:
      - label: 'Home'
        name: 'home'
        file: 'content/horses.md'
        fields:
          - { label: 'Title', name: 'title', widget: 'string' }
          - { label: 'Publish Date', name: 'date', widget: 'datetime' }
          - { label: 'Body', name: 'body', widget: 'markdown' }
          - label: 'Horses'
            name: 'horses'
            widget: list
            fields:
              - { label: 'Name', name: 'name', widget: 'string' }
              - { label: 'Description', name: 'description', widget: 'text' }
              - { label: 'Character', name: 'character', widget: 'text' }

```

content/horses.md



```
---
title: Great Horses
date: 2021-02-22T23:31:20:551Z
horses:
  - name: Excepzional
    description: Excepzional a grey Polish-Crabbet Arabian stallion and an experienced competitor.
    character: Intelligent, opinionated, FAST. Wants to win.
  - name: Chanz
    description: Chanz is a chestnut colored Arabian out of the Rushcreek line.
    character: Great sense of humor, curious, removes bucket lids, easily spooked, challenge to stay in saddle.
  - name: Aris
    description: Aris is a grey Arabian and half brother to Chanz.
    character: Earnest, cautious. Hell of a bucking horse (unfortunately!).
---

This is the profile page for some of my horses.

Also, this features Next.js as a React.js framework

```

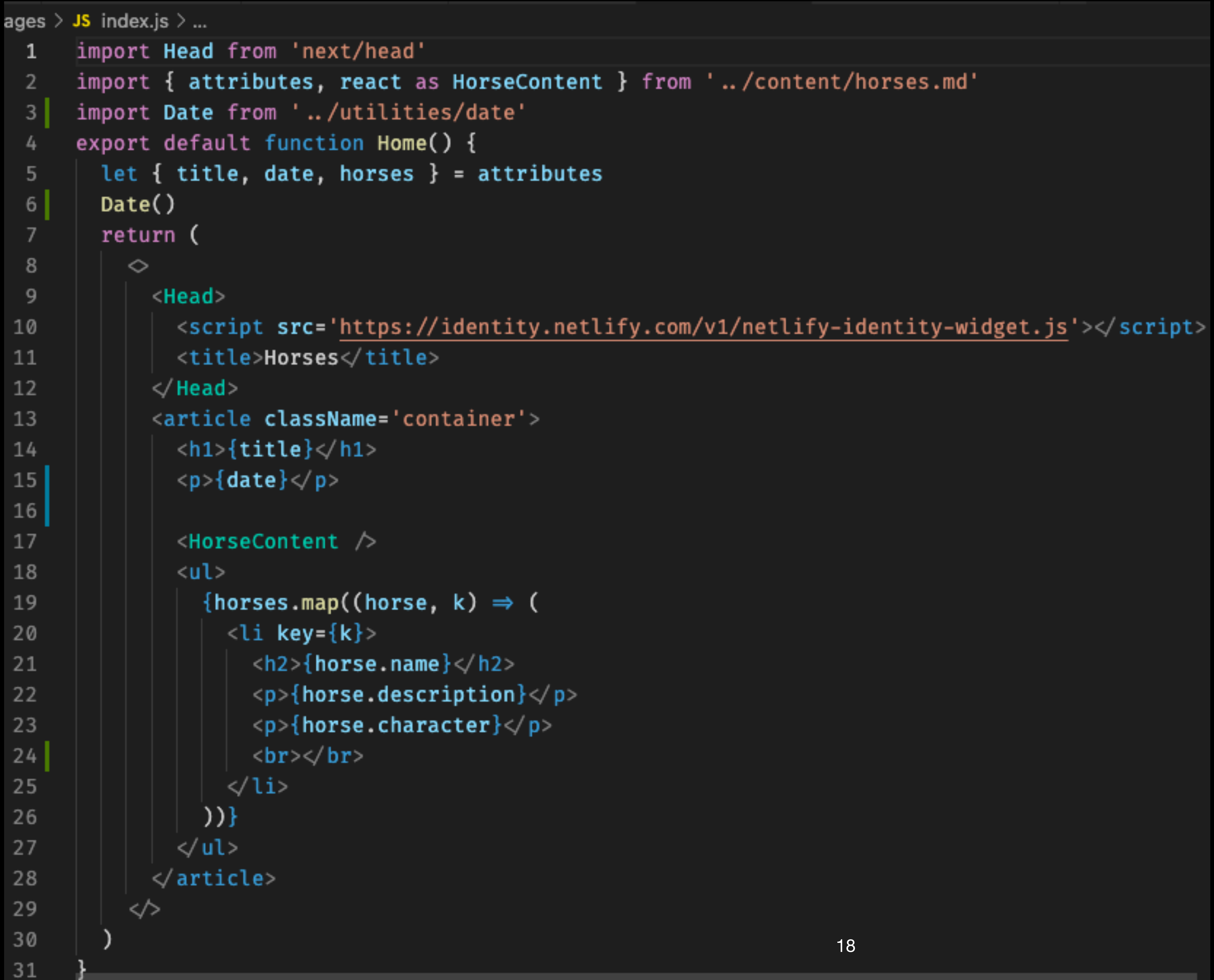
next.config.js - create in project root



next.config.js > ...

```js
module.exports = {
  webpack: (cfg) ⇒ {
    cfg.module.rules.push({
      test: /\.md$/,
      loader: 'frontmatter-markdown-loader',
      options: { mode: ['react-component'] },
      // target: 'serverless',
    })
    return cfg
  },
}
```

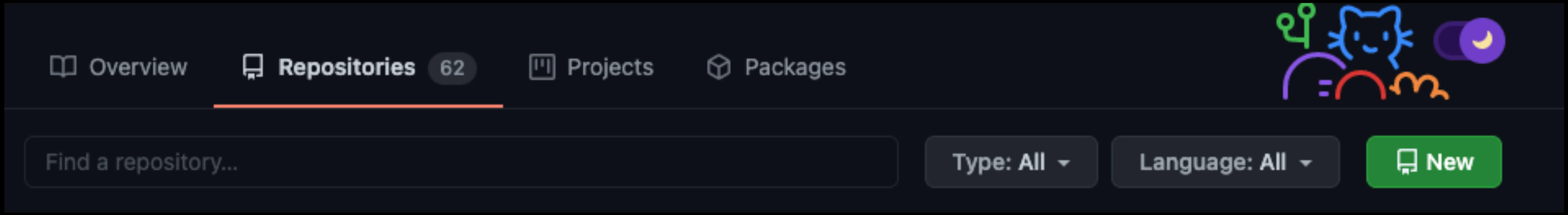
pages/index.js - replace content



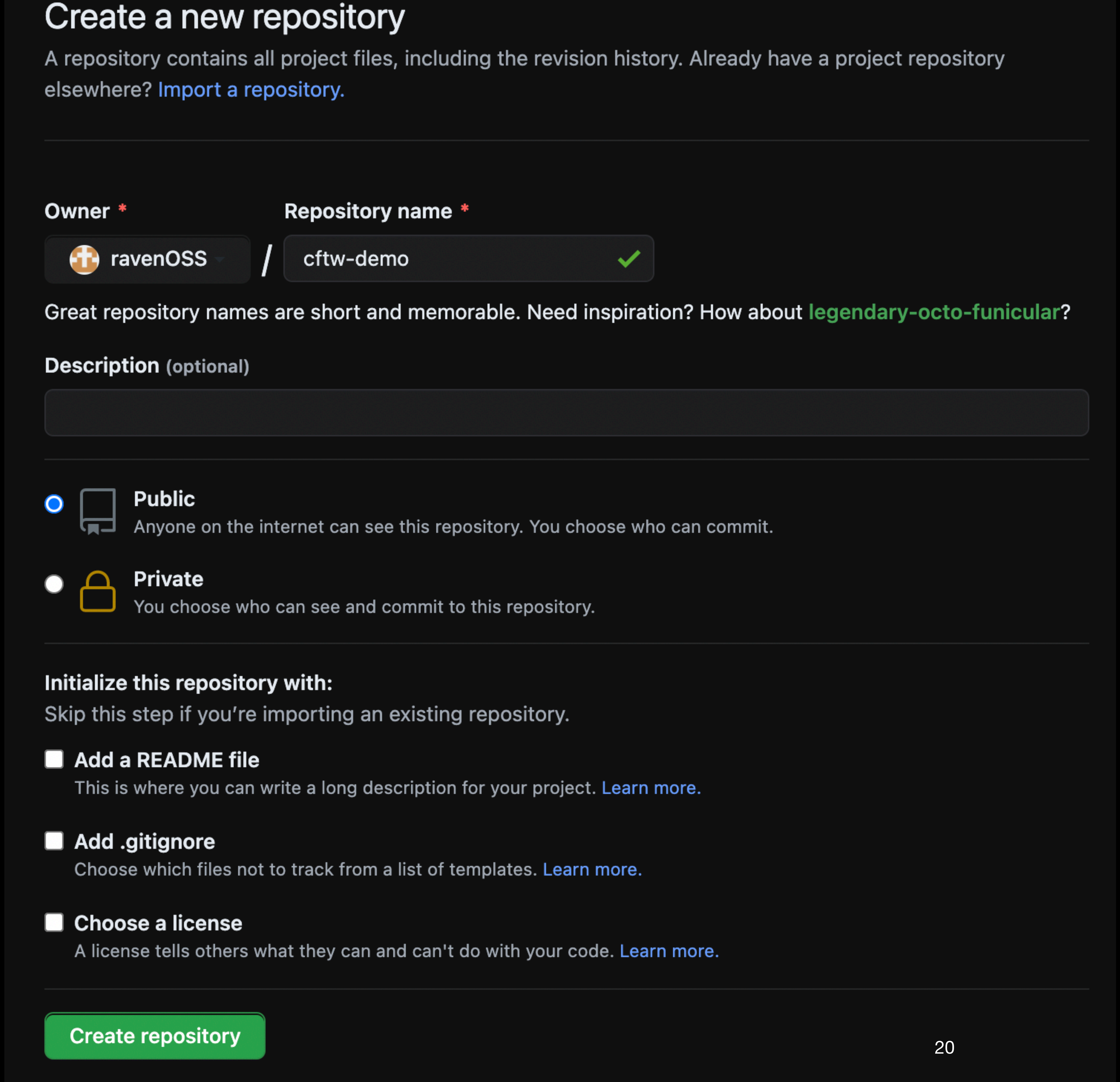
```jsx
import Head from 'next/head'
import { attributes, react as HorseContent } from '../content/horses.md'
import Date from '../utilities/date'
export default function Home() {
  let { title, date, horses } = attributes
  Date()
  return (
    <>
      <Head>
        <script src='https://identity.netlify.com/v1/netlify-identity-widget.js'></script>
        <title>Horses</title>
      </Head>
      <article className='container'>
        <h1>{title}</h1>
        <p>{date}</p>

        <HorseContent />
        <ul>
          {horses.map((horse, k) => (
            <li key={k}>
              <h2>{horse.name}</h2>
              <p>{horse.description}</p>
              <p>{horse.character}</p>
              <br></br>
            </li>
          ))}
        </ul>
      </article>
    </>
  )
}
```

Github

- A Github account is necessary because live deploys draw data from your repo
- Login or Create account
- Create a new repo by clicking green ‘New” button



  - Choose a name for the your repo
  - DO NOT add gitignore, readme, or license files
  - You should have the view shown on page 20



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * **Repository name** *

ravenOSS / cftw-demo ✓

Great repository names are short and memorable. Need inspiration? How about legendary-octo-funicular?

Description (optional)

● **Public**
Anyone on the internet can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

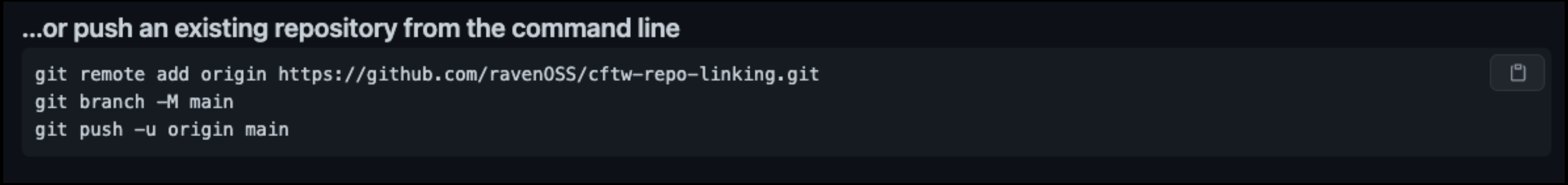
□ **Add a README file**
This is where you can write a long description for your project. Learn more.

□ **Add .gitignore**
Choose which files not to track from a list of templates. Learn more.

□ **Choose a license**
A license tells others what they can and can't do with your code. Learn more.

Push local repo

- Once a new Github repo is created, a screen is displayed with options. Choose the following option (from your repo):



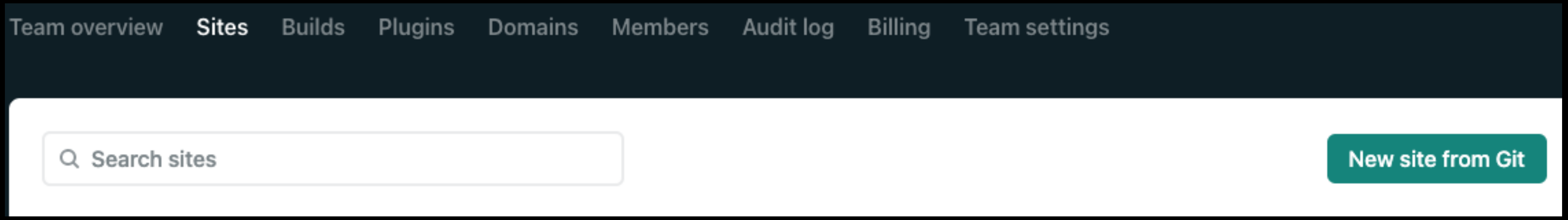
...or push an existing repository from the command line

```
git remote add origin https://github.com/ravenOSS/cftw-repo-linking.git
git branch -M main
git push -u origin main
```

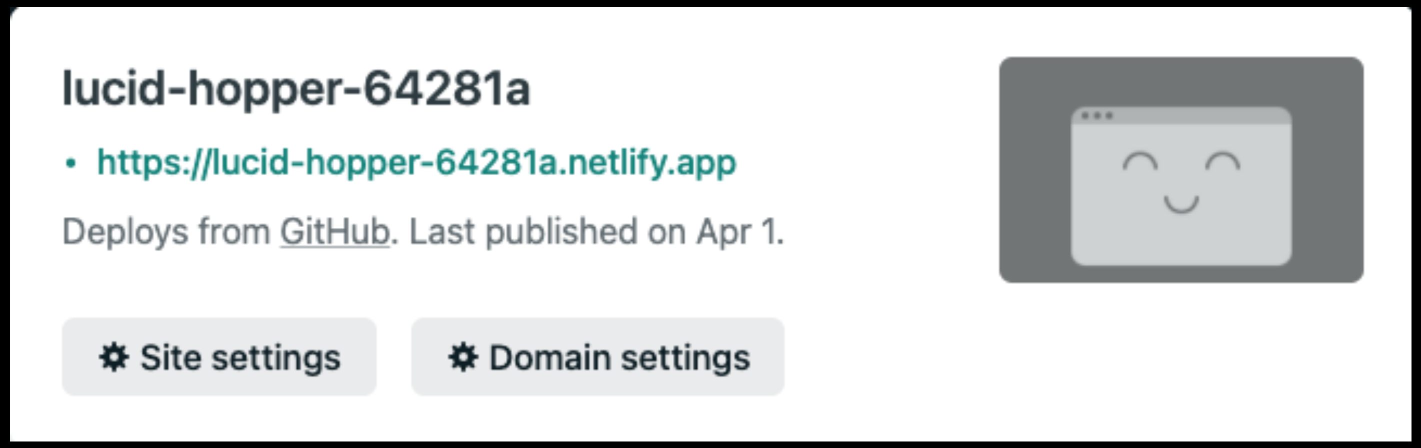
  - Copy all to clipboard and paste into the command line (terminal) within your project folder
  - If you get an error like ‘git not initialized, use these commands:
    - git add . (git add period)
    - git push -u origin main
  - You can visit your github repo and check for file additions

netlify.com

- Create Account
- New site from git



- Select your newly created Github repo(Netlify creates new site with random name)



lucid-hopper-64281a

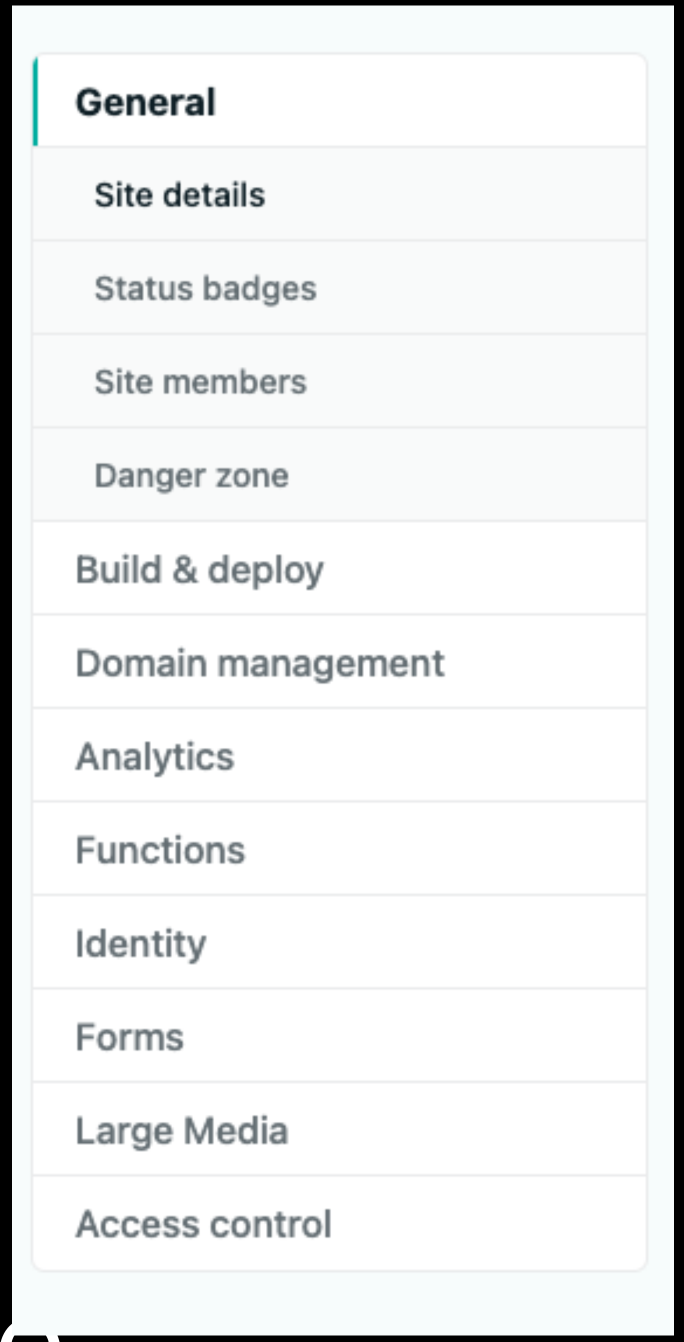
• https://lucid-hopper-64281a.netlify.app

Deploys from GitHub. Last published on Apr 1.

-

Site Settings - General Menu

- Site Settings > Identity > Registration > Registration Preferences > Edit Settings > Open
- External providers > select method (Github)
- Authenticate with your Github account



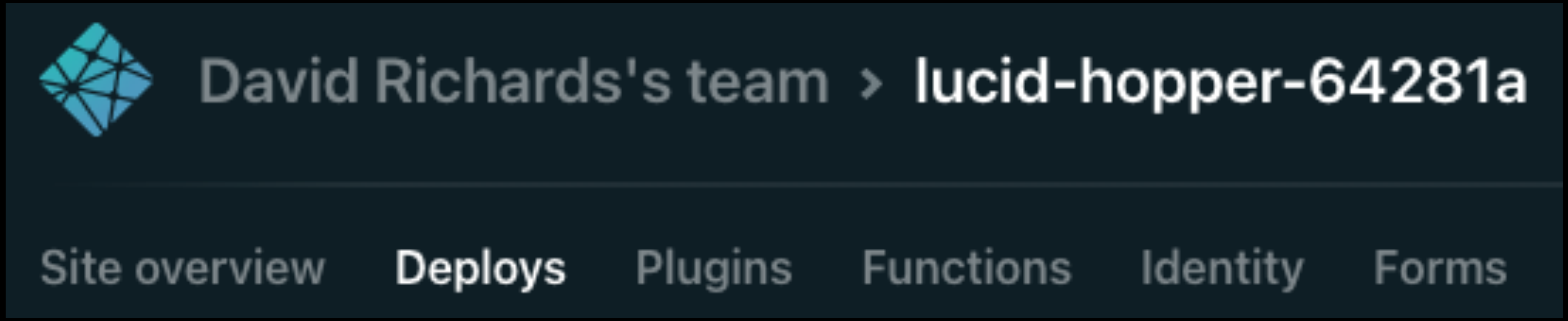
- Site Settings > Identity > Services > Git Gateway > Enable

Deploy Settings

- Site Settings > Build & Deploy > Build Settings
  - Repository: check name of your Github repo
  - Build: next build && next export
  - Publish directory: out
- Site Settings > Build & Deploy > Deploy Context
  - Production branch: probably Main or Master (CHECK GITHUB REPO FOR BRANCH)

Create Deployment

- Choose deploys from navigation (image below)
- If site is not already showing a deployment message (or failure), use “Trigger Deploy”.
- Wait a minute or two for the site to build and it will become available at site URL after deploy success message



Deployed!

- Note URL from deploys page (click on URL)
- Add “/admin” to URL in browser - provides Netlify Login
- Complete login USING GITHUB
- View CMS
- Workflow
  - Add horse to list > save (top left)
  - Top right of page > Status pending review (git staged)
  - New feature: A preview mode is now provided
  - Status pending > status > ready (git commit)
  - Publish (live)
  - NOTE: online changes DO NOT sync to local project - use ‘git pull’ to sync

Summary

- Create Next app - scaffolds basic configuration including Webpack
  - npx create-next-app <example> - generates specific config
  - Find examples in Nextjs Github repo
- Configure Netlify-cms with config.yml & index.html
  - Use editor mode to stage changes and prevent auto deploy (build minutes!)
- API routes - emulate node server PROVIDES BACKEND FUNCTIONS
  - Do not appear in frontend code
  - Enables protection of API keys
- Simple sites do not require a CMS! - Many hosting options

Additional exercises

Styling Options

- Useful exercise for learning about styling in Next.js.
- https://www.smashingmagazine.com/2020/09/comparison-styling-methods-next-js/
  - Examine comments at the end of article (I have provided notes to correct the article)

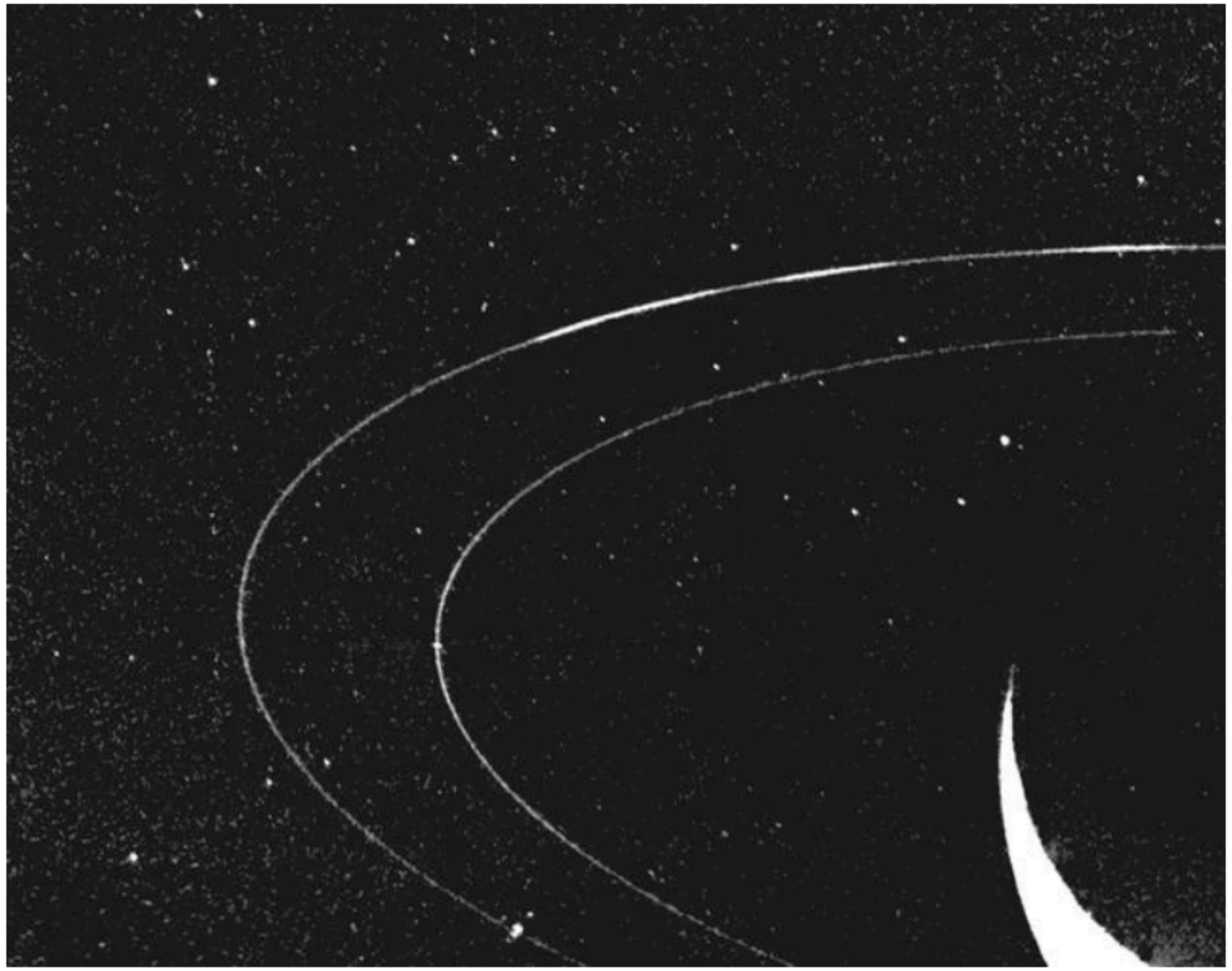
Static & Server Rendered Data

NASA - Demo static and server rendered

- https://github.com/cassidoo/next-nasa-demo/blob/master/pages/index.js
- https://next-nasa-demo.netlify.app/

NASA Demo

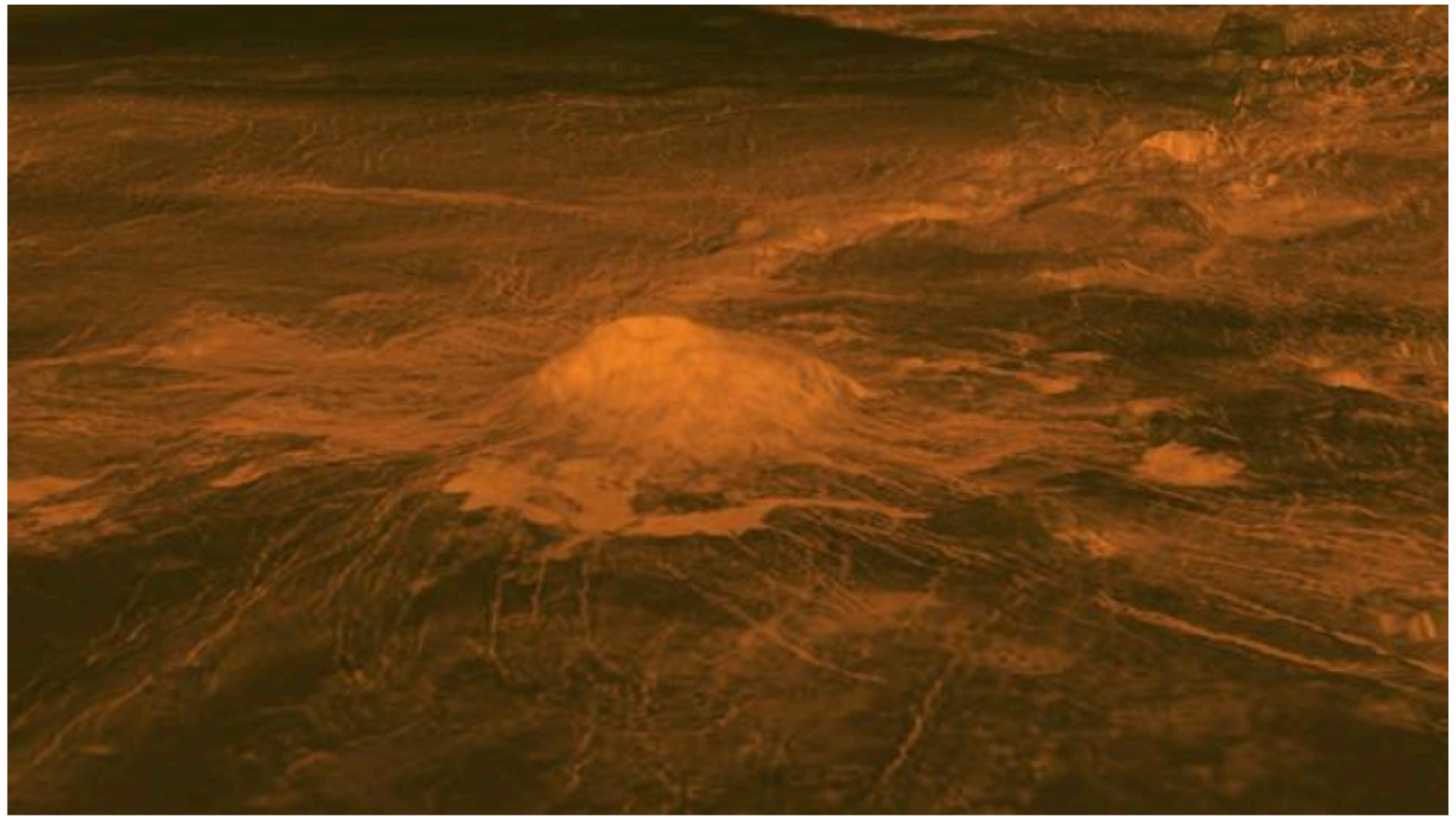
This image was fetched and rendered at build time.



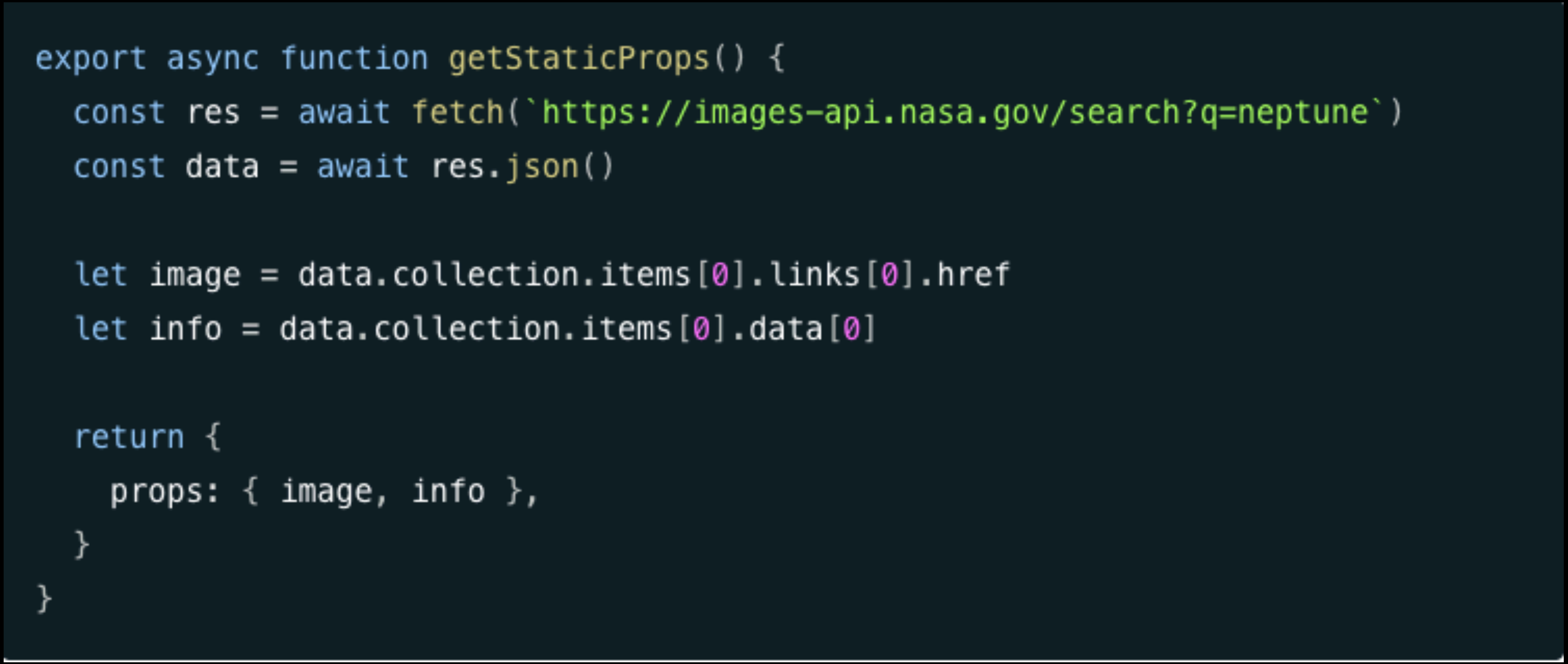
Neptune Rings: In Neptune outermost ring, 39,000 miles out, material mysteriously clumps into three arcs. Voyager 2 acquired this image as it encountered Neptune in August of 1989.

This lets you query an image at runtime.

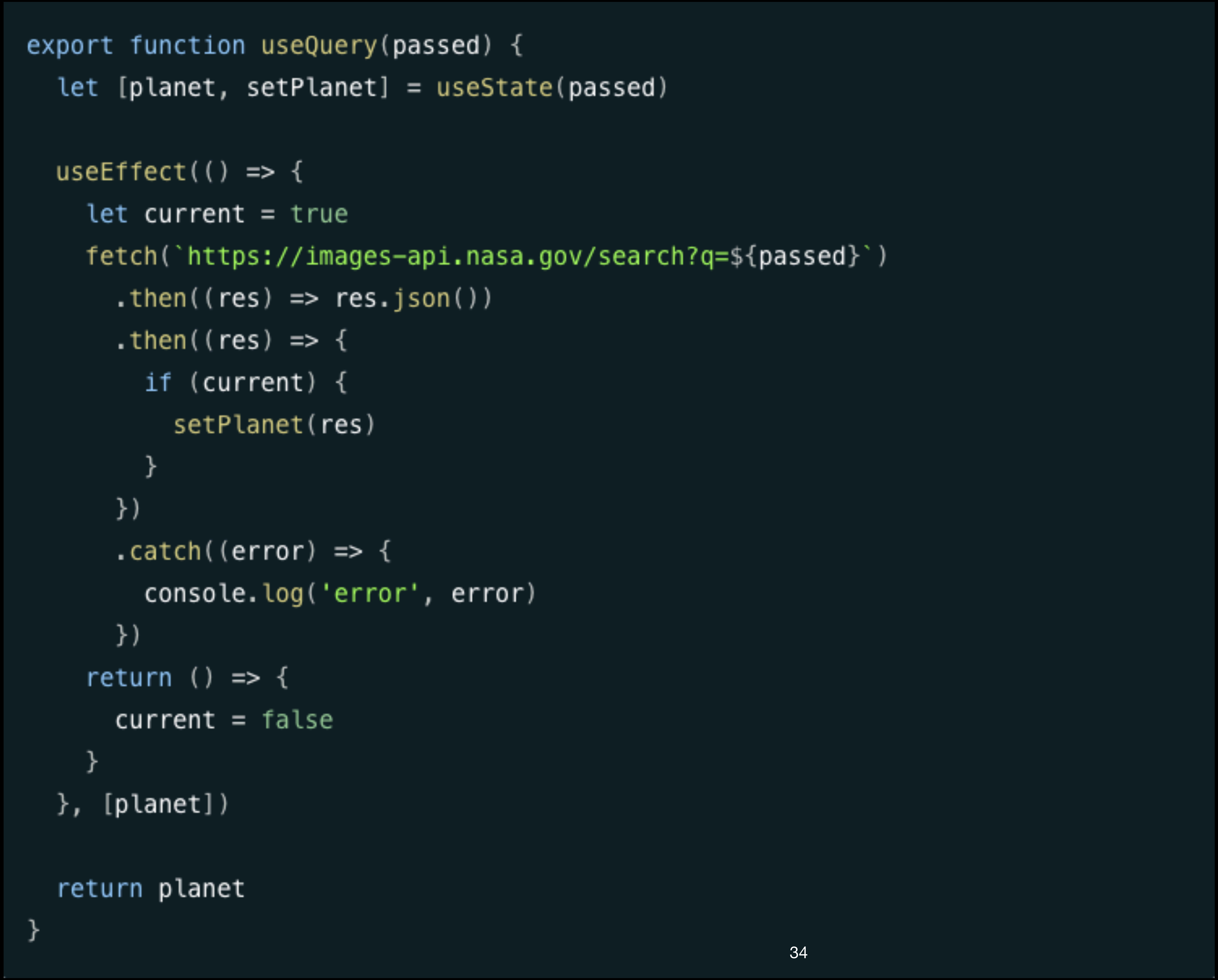
venus



Surface Warmth on a Venus Volcano: This figure shows the volcanic peak Idunn Mons in the Imdr Regio area of Venus. The topographic backbone brown color was derived from data obtained by NASA Magellan spacecraft and the overlay was derived from data from ESA Venus Express Spacecraft.



```
export async function getStaticProps() {
  const res = await fetch(`https://images-api.nasa.gov/search?q=neptune`)
  const data = await res.json()

  let image = data.collection.items[0].links[0].href
  let info = data.collection.items[0].data[0]

  return {
    props: { image, info },
  }
}
```



```
export function useQuery(passed) {
  let [planet, setPlanet] = useState(passed)

  useEffect(() => {
    let current = true
    fetch(`https://images-api.nasa.gov/search?q=${passed}`)
      .then((res) => res.json())
      .then((res) => {
        if (current) {
          setPlanet(res)
        }
      })
      .catch((error) => {
        console.log('error', error)
      })
    return () => {
      current = false
    }
  }, [planet])

  return planet
}
```