EXTERNAL DESIGN SYSTEMS IN PRACTICE

Brian Perry

June 7, 2019

Twin Cities Drupal Camp

Slides: http://bit.ly/externalds-tcd

bounteous

BRIAN PERRY

- Lead Front End Dev at Bounteous
- Rocking the Chicago ‘burbs
- Lover of all things components…

  …and Nintendo
- Increasingly JAMStack Curious

d.o: brianperry

twitter: bricomedy

github: backlineint

nintendo: wabrian

brianperryinteractive.com

bounteous

Mountain States
Health Alliance

fitbit

vsp
Vision care for life

zipcar

Allstate
You're in good hands.

EyeMed
Vision Care

Centura
Health

FEDERAL-MOGUL
MOTORPARTS

P&G

CHS
Community
Health Systems

AON
Hewitt

Wilson

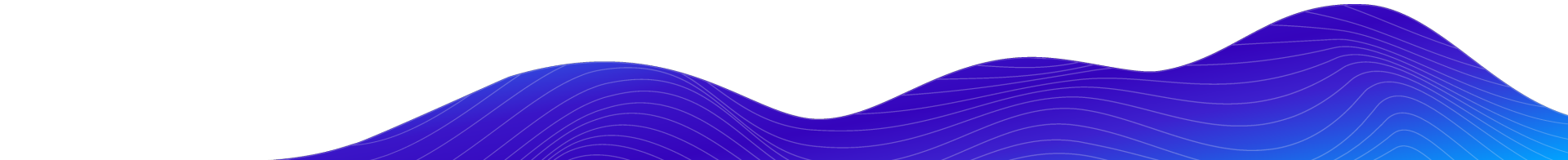
spring

invaluable

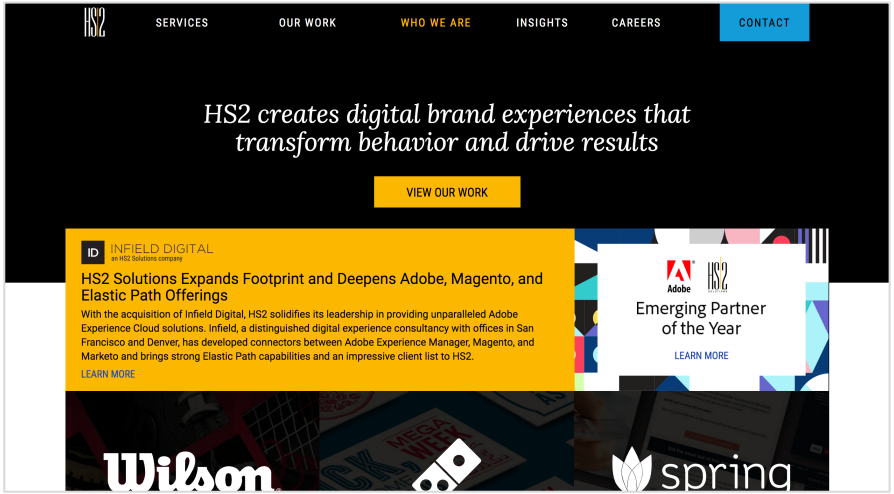
HS2 SOLUTIONS

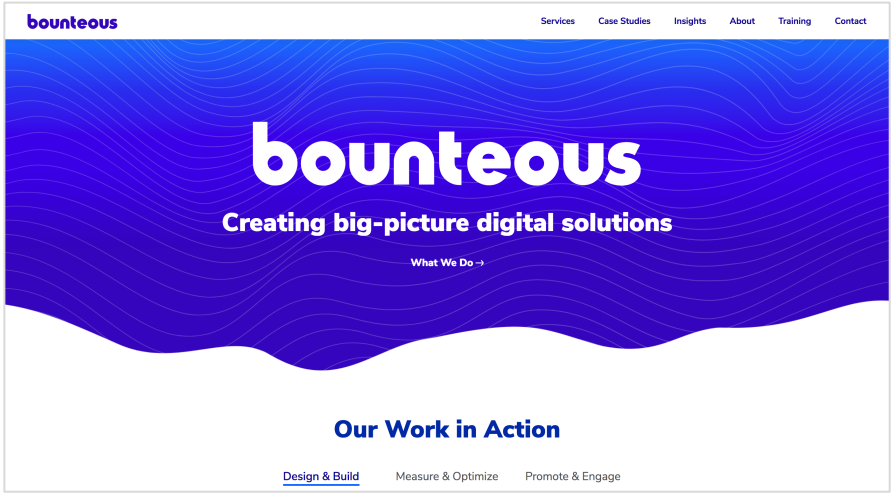
A REBRANDING STORY

or: How I Fell Into Using an External Design System



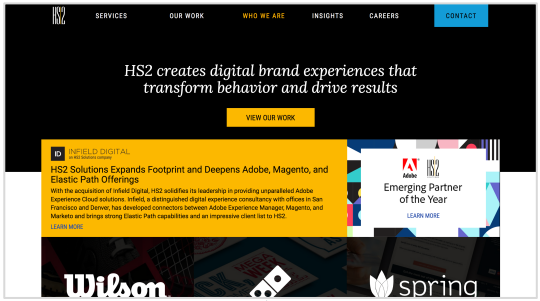
REBRANDING

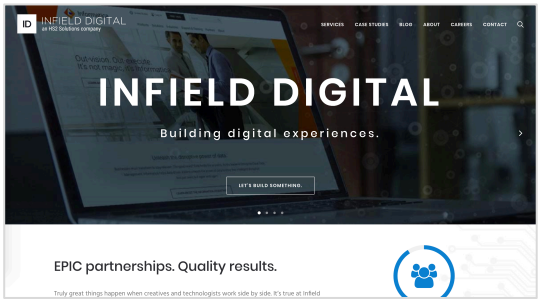


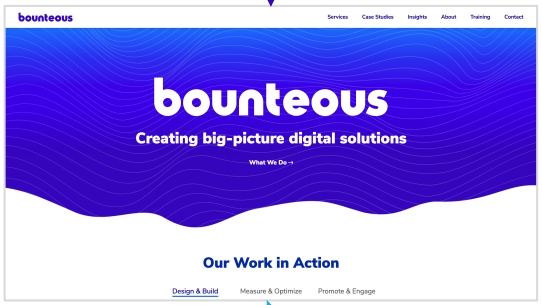


- Brand launch targeted for November 5, 2018
- Website is only one part in a larger coordinated effort

MIGRATION

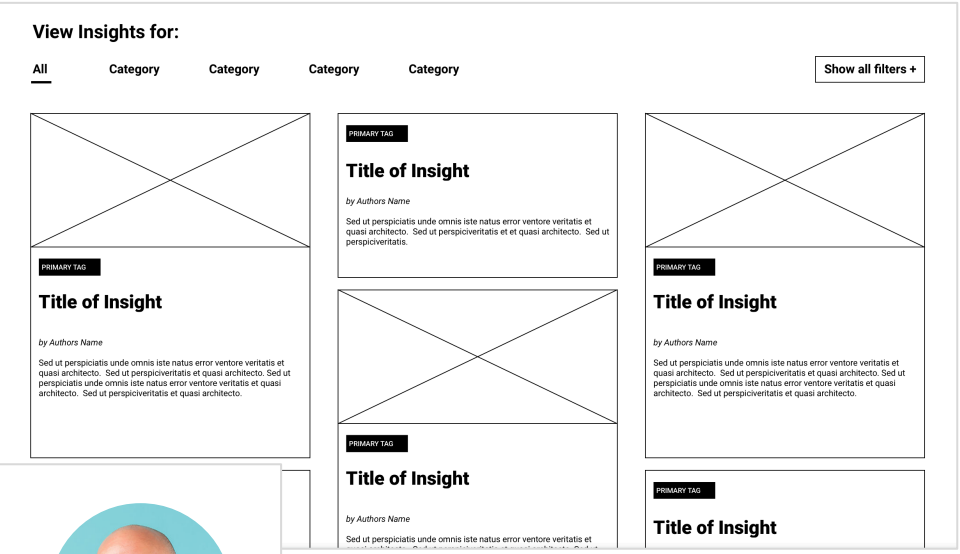


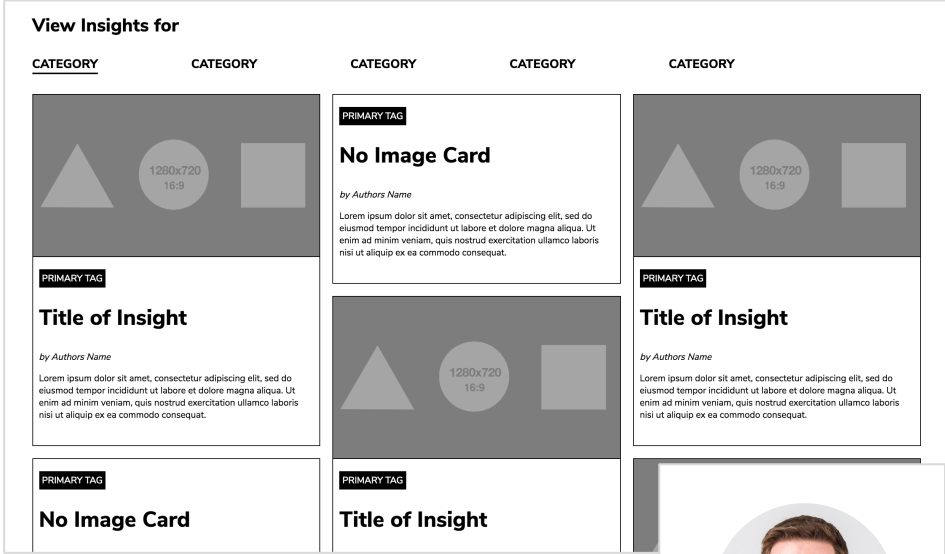




Nate
Dev

MEANWHILE...



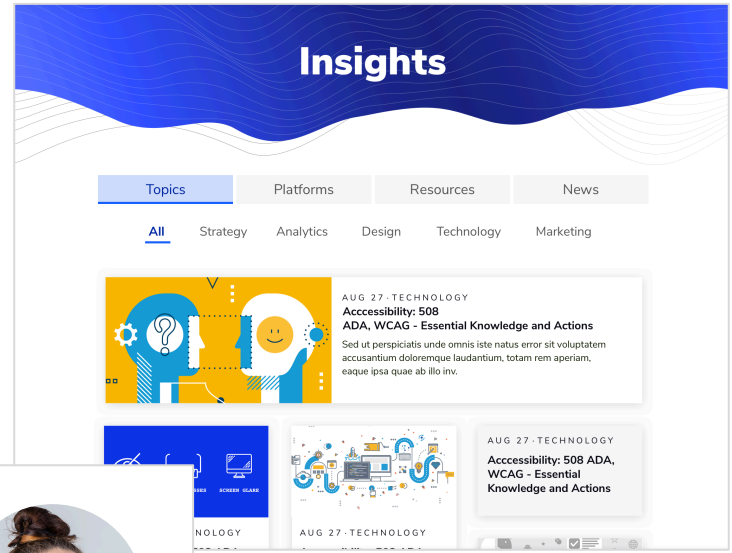


Leon
XD

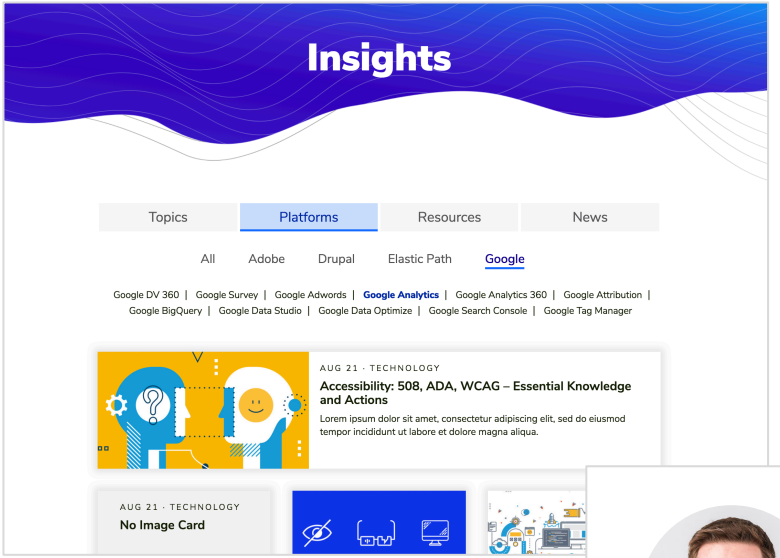
pattern lab

Brian
FE Dev

PROTOTYPING



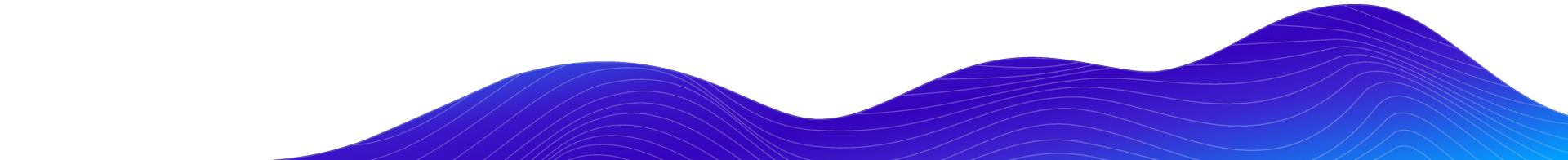
Joleen
UX



Brian
FE Dev

DRUPAL INTEGRATION

or: Deciding Where To Put The Mustard



VS

Past projects: Pattern Lab embedded in theme

- Easier for Drupal to access components
- Simplified build process
- Easier for developers to understand
- Prone to Drupalisms leaking into components

Current state: Independent Projects

- Drupal Repository
- Design System Repository

Promotes use beyond a single Drupal project.

Where should we put the mustard?

LET'S KEEP THE MUSTARD SEPARATE

- Keeps components Drupal agnostic
- Continues to encourage prototyping
- Already has been living as a separate project
- Some gut feelings:
  - **Conceptually feels like the right thing**
  - Advocated by many in the community who I trust
  - May be able to re-use on other web projects related to our brand.
  - Feels right for a company with diverse skills

REQUIRING THE DESIGN SYSTEM

composer.json – in design system

```
{
  "name": "hs2studio/hs2solutionsdesignsystem",
  "description": "HS2 Solutions Design System",
  "type": "design-system"
}
```

REQUIRING THE DESIGN SYSTEM

composer.json – in Drupal project

Add Design system repository

```
12          "repositories": [
13              {
14                  "type": "composer",
15                  "url": "https://packages.drupal.org/8"
16              },
17              {
18                  "type": "git",
19                  "url": "git@bitbucket.org:hs2studio/hs2solutionsdesignsystem.git"
20              },
```

REQUIRING THE DESIGN SYSTEM

composer.json – in Drupal project

Define how to handle design-system installer types

```
153         "extra": {
154             "installer-types": [
155                 "design-system"
156             ],
157             "installer-paths": {
158                 "docroot/core": ["type:drupal-core"],
159                 "docroot/libraries/{$name}": ["type:drupal-library"],
160                 "docroot/themes/custom/hsnew/libraries/{$name}": ["type:design-system"],
161                 "docroot/modules/contrib/{$name}": ["type:drupal-module"],
162                 "docroot/profiles/contrib/{$name}": ["type:drupal-profile"],
163                 "docroot/themes/contrib/{$name}": ["type:drupal-theme"],
164                 "drush/contrib/{$name}": ["type:drupal-drush"]
165             },
```

REQUIRING THE DESIGN SYSTEM

```
composer require hs2studio/hs2solutionsdesignsystem
```

- Required dev for pre-launch cycle

Updating

```
composer update hs2studio/hs2solutionsdesignsystem
```

- Releases are tagged in design system
- Can require design system at a specific release

INCORPORATING INTO THEME

hsnew.libraries.yml

Global library with compiled assets

```
global:
  version: 1.x
  css:
    theme:
      libraries/hs2solutionsdesignsystem/source/dist/style.css: {}
      css/drupal.css: {}
      "https://fonts.googleapis.com/css?family=Nunito+Sans:400,600,700,800,900": { type:
  js:
    libraries/hs2solutionsdesignsystem/source/dist/base.bundle.js: {}
    libraries/hs2solutionsdesignsystem/source/dist/jquery.bundle.js: {}
  dependencies:
    - core/drupal
    - core/jquery
    - core/jquery.once
```

INCORPORATING INTO THEME

hsnew.info.yml

Twig namespaces for ‘atomic’ component categories

```
32          component-libraries:
33            images:
34              paths:
35                - libraries/hs2solutionsdesignsystem/source/images
36            elements:
37              paths:
38                - libraries/hs2solutionsdesignsystem/source/_patterns/elements
39            components:
40              paths:
41                - libraries/hs2solutionsdesignsystem/source/_patterns/components
42            regions:
43              paths:
44                - libraries/hs2solutionsdesignsystem/source/_patterns/regions
45            layouts:
46                paths:
47                  - libraries/hs2solutionsdesignsystem/source/_patterns/layouts
```

COMPONENT WORKFLOW

Sean
Design System

Wade
Drupal

Design System Components

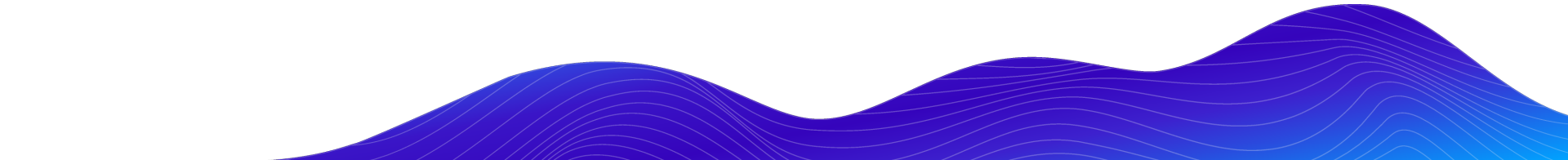
- Created the bulk of design system components, primarily outside of Drupal.
    - Only 2 commits in Drupal repo

Drupal Component Integration

- Wired design system components to Drupal content and layouts.
    - Majority of effort in Drupal repository
    - Minor design system refactoring when necessary

WE’RE GOING TO NEED MORE MUSTARD

or: How an External Design System Helped Us Handle Changes in Scope



LUNAMETRICS TRAINING SITE SCOPE

Training registration rebuild de-scoped

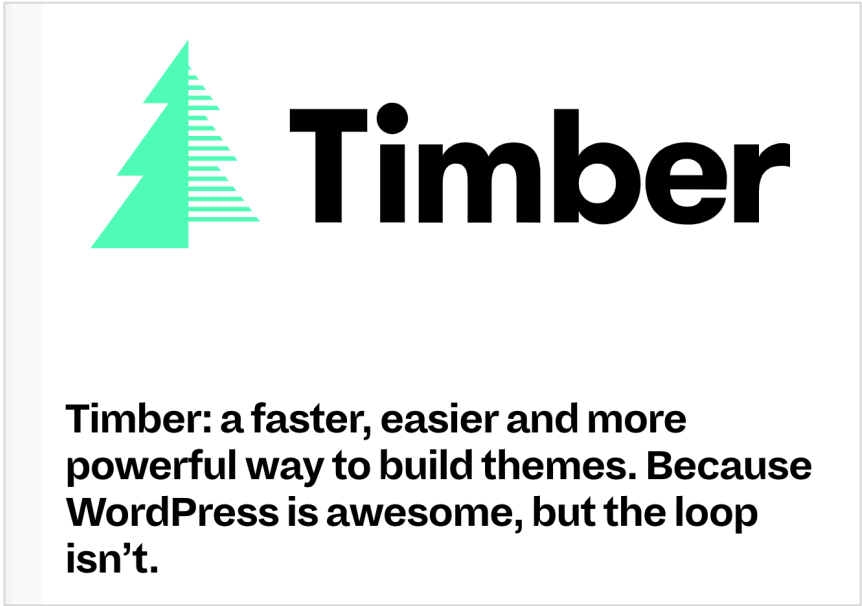
- Existing WordPress site lives on at training.bounteous.com subdomain.
- Needs to match look and feel of D8 site

This design system thing seems pretty handy now.

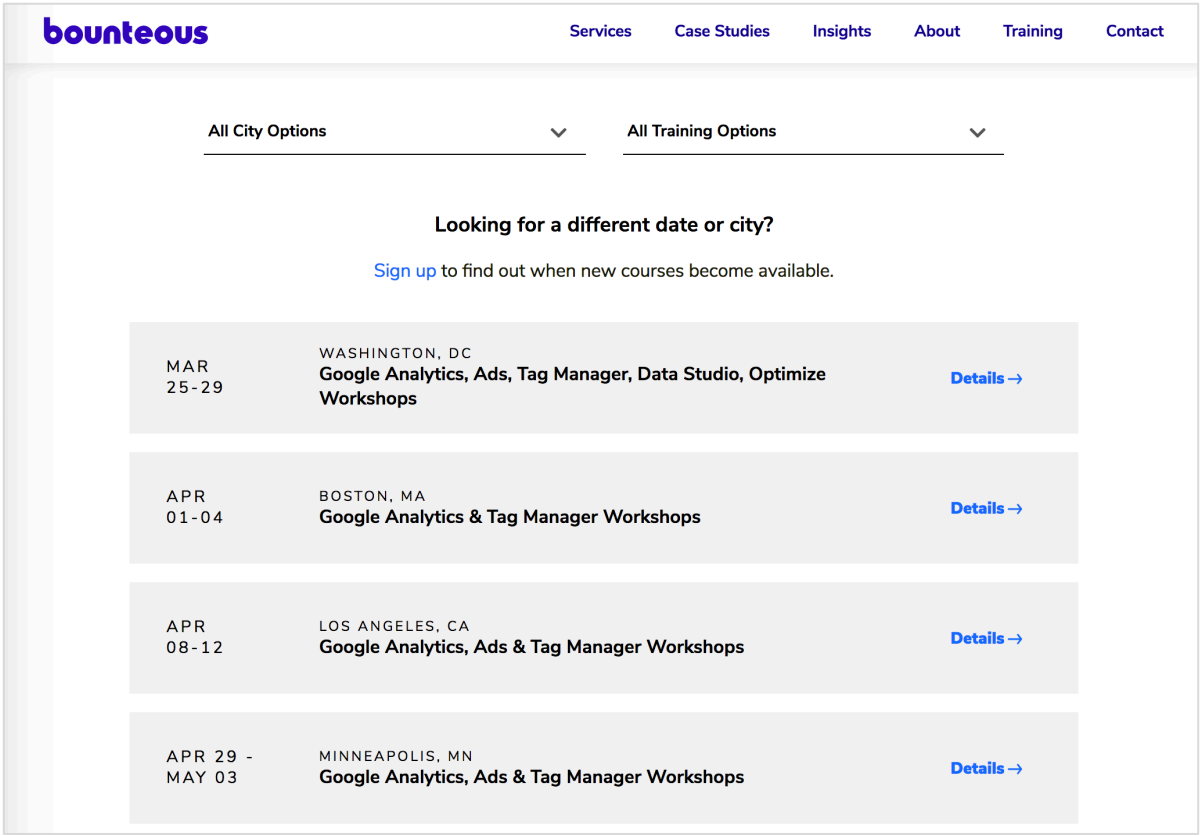
WORDPRESS INTEGRATION

Worked with Wahila Creative to re-theme the site.

- Introduced us to Timber plugin which makes it possible to use Twig in WordPress.
- Initial prototype used Timber and design system.



TIMBER COMPONENT – TRAINING ITEMS



TIMBER COMPONENT – TRAINING ITEMS

training-module.php

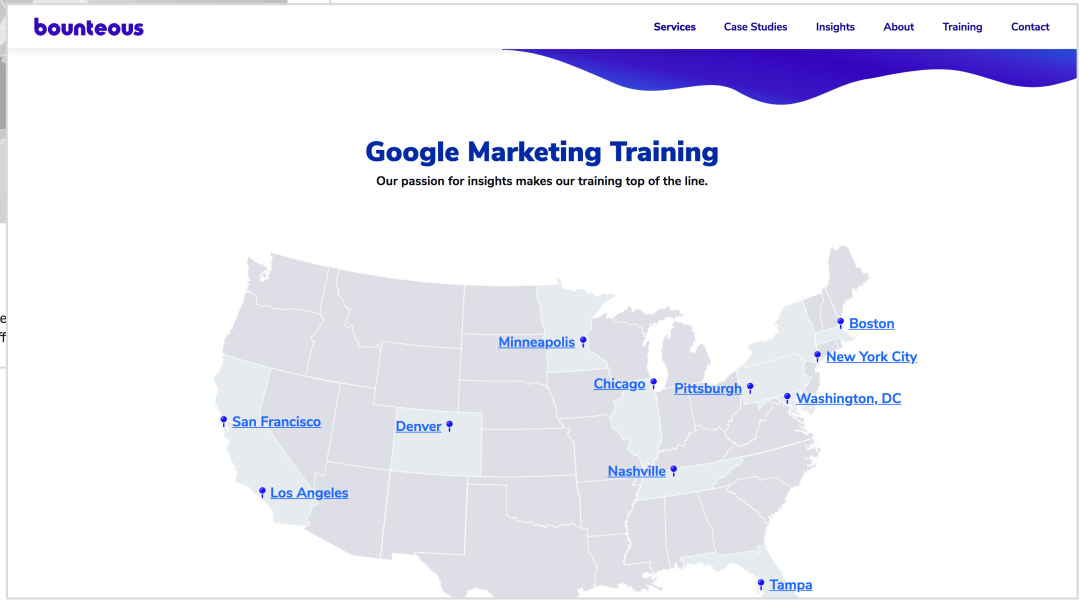
```
20    <div class="c-training-module__wrapper js-trainingList">
21      <?php
22        foreach ($data['courses'] as $event) :
23          $trainingElement([
24            'id'        => $event['eventbrite_id'],
25            'month'      => get_formated_date_string_months ( strtotime($event['start_date']),
26              strtotime($event['end_date']) ),
27            'days'      => get_formated_date_string_days ( strtotime($event['start_date']),
28              strtotime($event['end_date']) ),
29            'location' => $event['event_city'] . ', ' . $event['event_state'],
30            'title'     => $event['title'],
31            'cta_link'      => $event['parent'] . '/#' .
32              getFormattedMonth($event['start_timestamp'])
33          ]);
34        endforeach;
35      ?>
36    </div>
```

TIMBER COMPONENT – TRAINING ITEMS

training-element.php

```php
<?php
/**
 * @uses {Array} $data Scoped Array of data used to populate the template
 *   @key {String} datemonths
 *   @key {String} datedays
 *   @key {String} location
 *   @key {String} title
 *   @key {String} link
 */
?>

<?php

if (!empty($data)) {
  // Render the design system component
  Timber::render('e-training.twig', $data);
}

?>

```

LunaMetrics
an HS2 Solutions company
GA360
Services
Training
About
LunaLabs
Contact
Blog
1.877.220.LUNA
Minneapolis
Boston
New York City
Chicago
Pittsburgh
Washington, DC
San Francisco
Denver
Nashville
Los Angeles
Digital Marketing Training Courses
At LunaMetrics we're genuinely passionate about what we do. This makes us ideal consultants, but it also make
than to educate people and allow them to make informed decisions that will help their online presences. We off
number of digital marketing areas.


bounteous
Services
Case Studies
Insights
About
Training
Contact
Google Marketing Training
Our passion for insights makes our training top of the line.
Minneapolis
Boston
New York City
Chicago
Pittsburgh
Washington, DC
San Francisco
Denver
Nashville
Los Angeles
Tampa

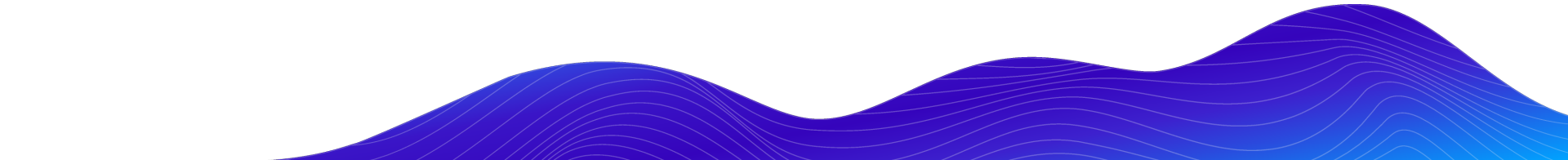
DEMAC MEDIA

Demac Media joins the family... very close to launch.

- Rebranded as bounteous.ca
- Smaller scale site, will differ from bounteous.com look and feel.
- More manual / selective use of design system assets.

DEMACMEDIA

POST LAUNCH



ONBOARDING

Kyle
Developer

Steve
Developer

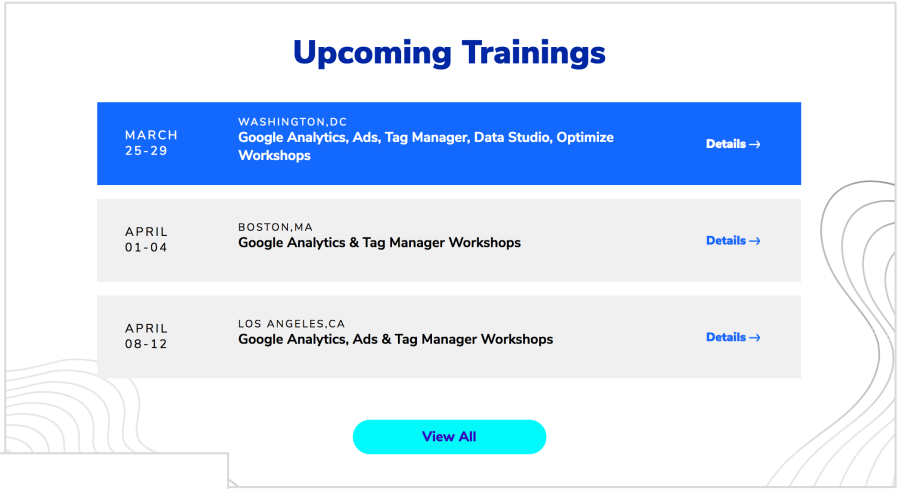
- Post launch, team members rotate.
- Majority of tickets are Drupal specific fixes. Faced with the design system they may wonder:

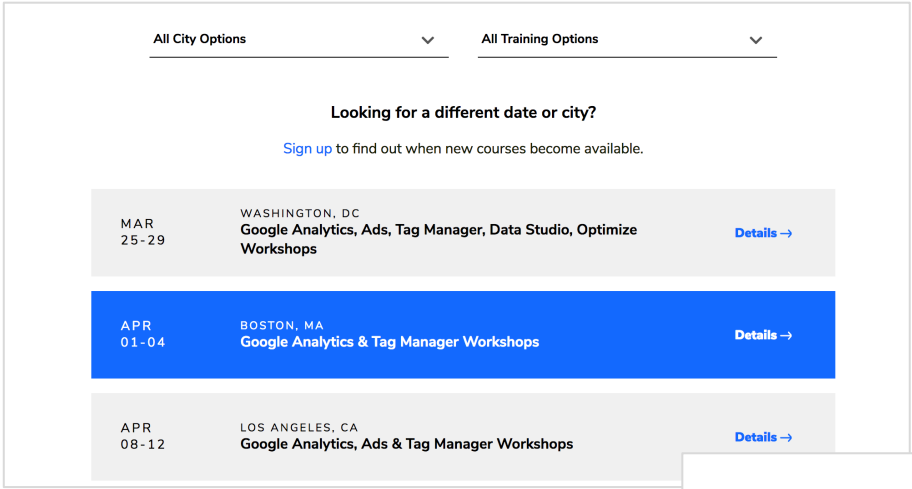
Why can’t I just Drupal?

Lessons Learned

- Documentation is important
  - Not just how, but why.
- Must have an efficient workflow for design system experimentation in CMS.
- Need to be strict about versioning.
- External approach can add overhead.

LINES BETWEEN CMSES BLUR





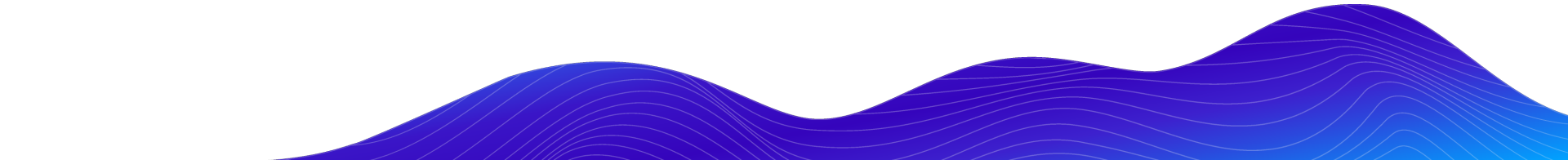
Kristen
QA

People will expect front-end changes to apply everywhere.

Jon
'Marketing'

DECOUPLED DESIGN SYSTEMS

Sharing between multiple templating engines

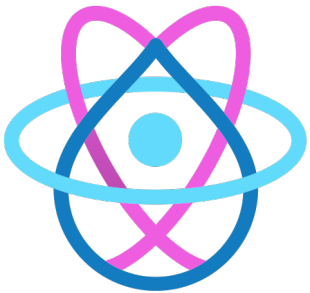


A PROGRESSIVELY DECOUPLED PROJECT

Drupal

- ‘Traditional’ Drupal Build
- Provides APIs to React

Dreact (DR8act?)

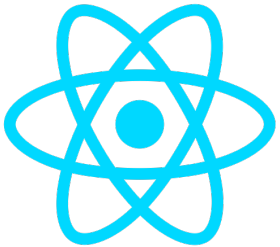


- Traditional Drupal Content

and

- Routes owned by React for User Portal

React



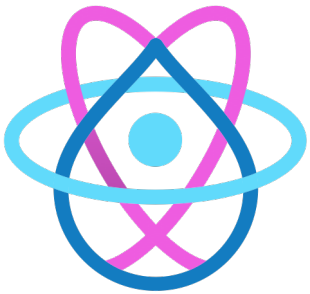
- Consumes Drupal APIs
- Authenticated User Portal

DX – MULTIPLE DEVELOPER CONTEXTS

Drupal

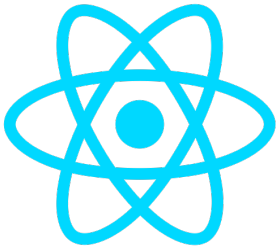
- Drupal implementation without React dependency.

Combined



- Test real world integration

React



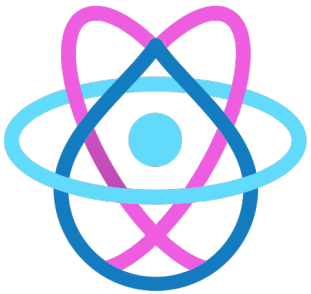
- Develop and test without local Drupal environment.
- Hot Module Replacement

TEMPLATING AND STYLES

Drupal

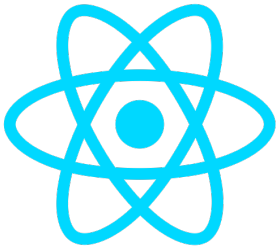
- Twig
- Unique styles and components

Combined



- Global styles that apply to both.
- Few (if any) shared components

React



- JSX
- Unique styles and components

COULD A DESIGN SYSTEM SERVE BOTH PROJECTS?

Drupal

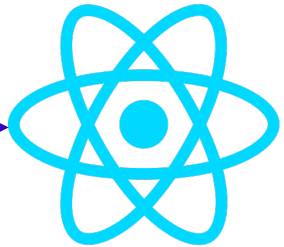
- Twig
- Unique styles and components

Design System

???

- **Global styles that apply to both.**

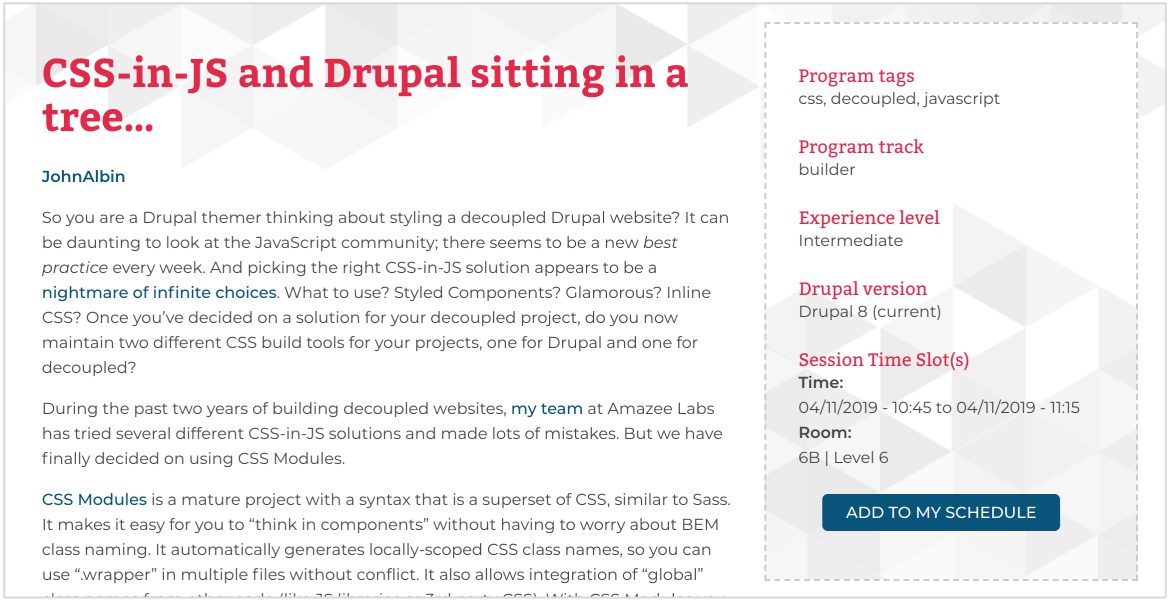
React



- JSX
- Unique styles and components

CSS IN JS IS ALL THE RAGE*

* occasional emphasis on RAGE



CSS-in-JS and Drupal sitting in a tree...

JohnAlbin

So you are a Drupal themer thinking about styling a decoupled Drupal website? It can be daunting to look at the JavaScript community; there seems to be a new *best practice* every week. And picking the right CSS-in-JS solution appears to be a nightmare of infinite choices. What to use? Styled Components? Glamorous? Inline CSS? Once you've decided on a solution for your decoupled project, do you now maintain two different CSS build tools for your projects, one for Drupal and one for decoupled?

During the past two years of building decoupled websites, my team at Amazee Labs has tried several different CSS-in-JS solutions and made lots of mistakes. But we have finally decided on using CSS Modules.

CSS Modules is a mature project with a syntax that is a superset of CSS, similar to Sass. It makes it easy for you to "think in components" without having to worry about BEM class naming. It automatically generates locally-scoped CSS class names, so you can use ".wrapper" in multiple files without conflict. It also allows integration of "global"

Program tags
css, decoupled, javascript

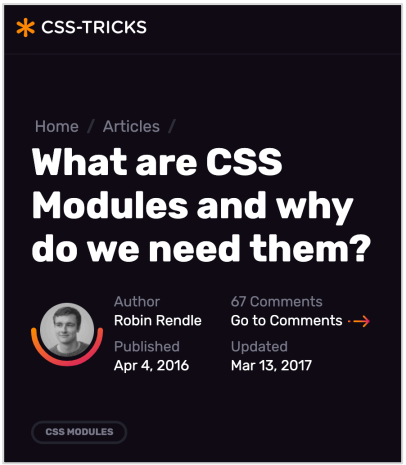
Program track
builder

Experience level
Intermediate

Drupal version
Drupal 8 (current)

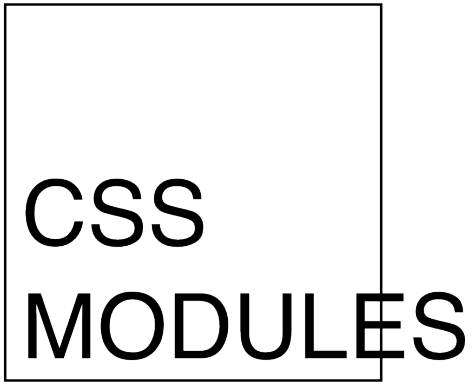
Session Time Slot(s)
Time:
04/11/2019 - 10:45 to 04/11/2019 - 11:15
Room:
6B | Level 6

ADD TO MY SCHEDULE

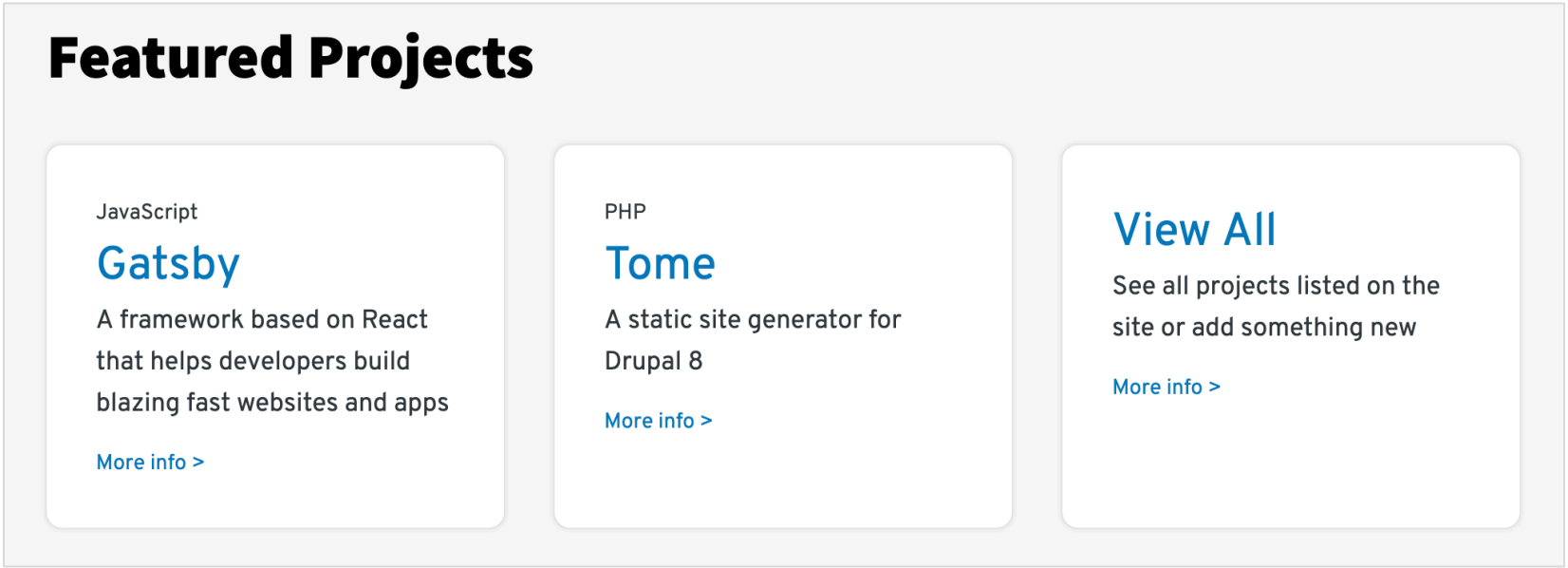


CSS MODULES

- Not a package
- Handled by a build process (typically Webpack)
- Scopes styles locally by default
- Can play nice with Sass
- Has a super snarky logo



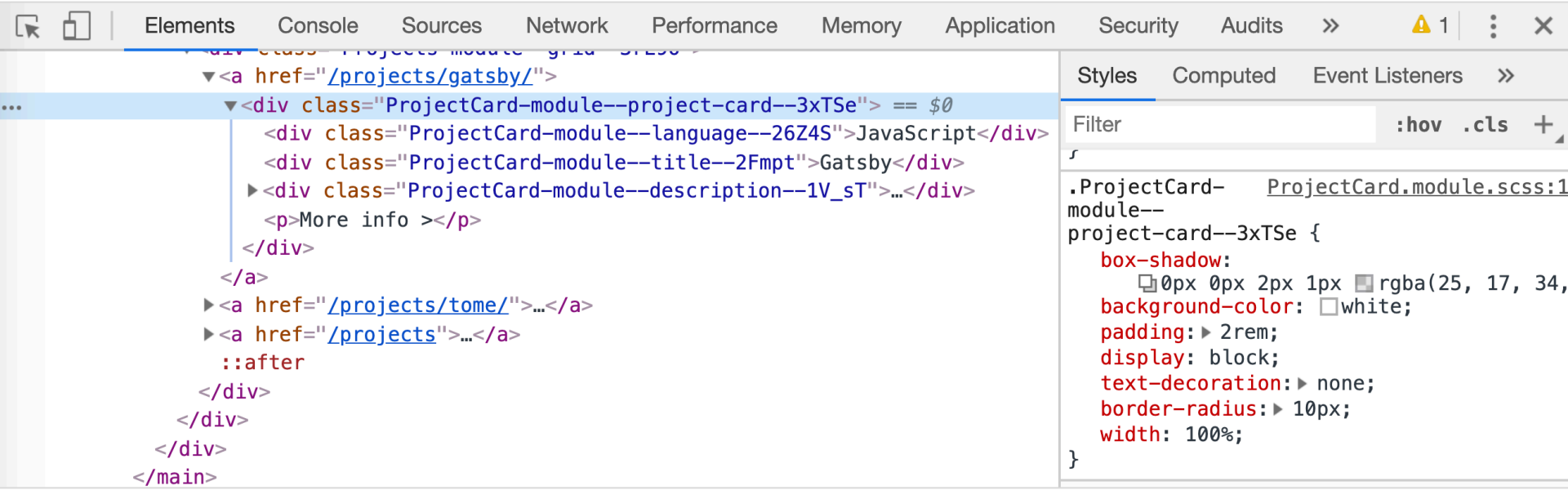
SCOPED CSS – PROJECT CARD



```scss
// ProjectCard.module.scss

.project-card {
  box-shadow: 0px 0px 2px 1px rgba(25,17,34,0.1);
  background-color: white;
  padding: 2rem;
  display: block;
  text-decoration: none;
  border-radius: 10px;
  width: 100%;

  &:hover {
    box-shadow: 0px 0px 2px 1px rgba(6, 120, 190, 0.3);
  }

  .title {
    font-family: 'Overpass', Sans-Serif;
    font-size: 30px;
  }

  .language {
    color: #31373f;
    font-size: 14px;
  }

  .description {
    color: #31373f;
  }

  p {
    font-size: 14px;
    margin-bottom: 0;
  }
}
```

```
// ProjectCard.js

import styles from "./ProjectCard.module.scss"

const ProjectCard = (props) => (
  <Link
    to={props.link}
    key={props.link}
  >
    <div className={styles.projectCard}>
      <div className={styles.language}>{props.language}</div>
      <div className={styles.title}>{props.title}</div>
      <div className={styles.description}>{props.description}</div>
      <p>More info ></p>
    </div>
  </Link>
);

export default ProjectCard
```

SCOPED CLASS NAMES

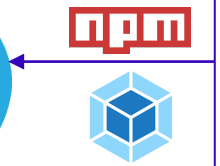


GLOBAL CSS

```
// ProjectCard.js

import React from "react"
import { Link } from "gatsby"

import "./ProjectCard.scss"

const ProjectCard = (props) => (
  <Link
    to={props.link}
    key={props.link}
  >
    <div className="project-card">
      <div className="language">{props.language}</div>
      <div className="title">{props.title}</div>
      <div className="description">{props.description}</div>
      <p>More info ></p>
    </div>
  </Link>
);

export default ProjectCard
```

DESIGN SYSTEM INFORMS BOTH PROJECTS

Drupal

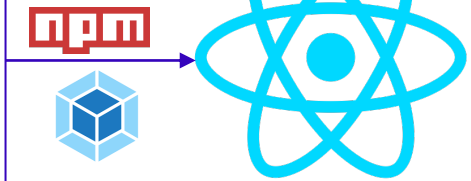


- Global CSS file - combines design system partials and Drupal specific partials

Design System

- Gatsby site to demonstrate global styles
- Projects consume partials as needed

React



- Imported in JS
- Mix of global and component scope
- Also project specific CSS

CONSUMING THE DESIGN SYSTEM

```
  },
  "devDependencies": {
    "my-design-system": "git@bitbucket.org:hs2studio/my-design-system.git#v1.0.3"
  }
}
```

- Use ~ for non-relative imports

```
/* Base
   ========================================================================== */
@import '~my-design-system/ds-styles/base/a11y';
@import '~my-design-system/ds-styles/base/base';
@import '~my-design-system/ds-styles/base/block';
@import '~my-design-system/ds-styles/base/buttons';
@import '~my-design-system/ds-styles/base/forms';
@import '~my-design-system/ds-styles/base/helpers';
@import '~my-design-system/ds-styles/base/style-guide';
@import '~my-design-system/ds-styles/base/typography';
@import '~my-design-system/ds-styles/base/vertical-rhythm';
@import 'base/drupal';

/* Element
   ========================================================================== */
@import '../../source/_patterns/02-elements/**/*.scss';

/* Component
   ========================================================================== */
@import '../../source/_patterns/03-components/**/*.scss';
```

WHAT WE AREN'T DOING (YET)

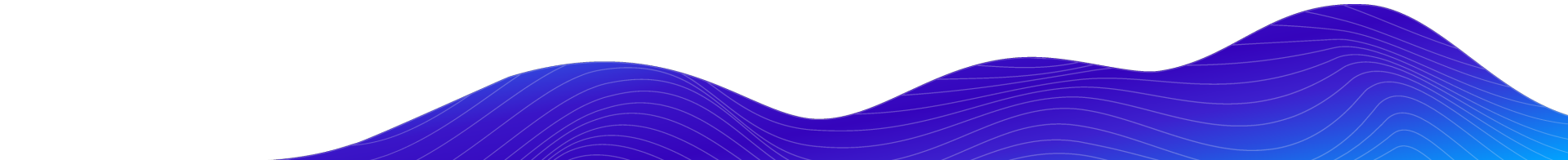
- Sharing Components
  - Limited overlap thus far
  - Probably requires more js – React or Web Components.
- Hot Module Replacement in Drupal

LESSONS LEARNED

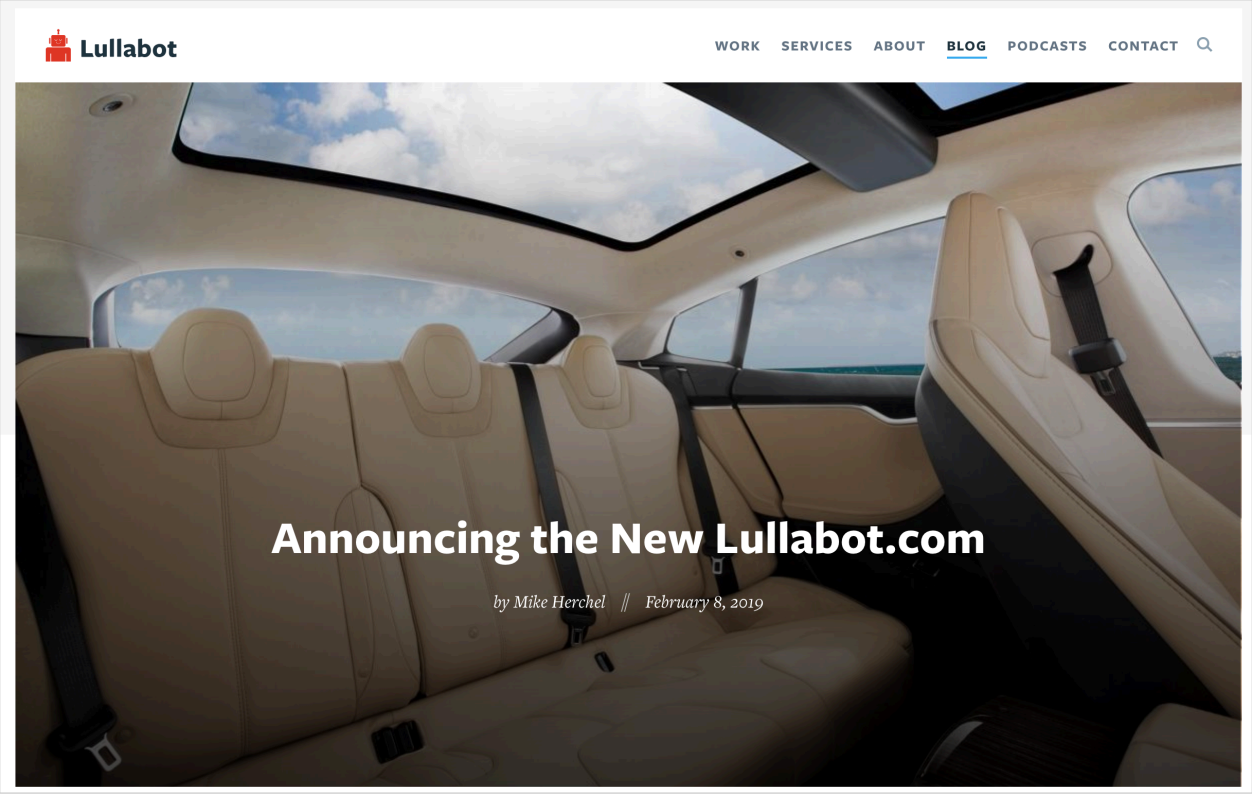
- Different Webpack build quirks can be a pain.
- Versioning continues to be important
  - Additional automation could help here.
- Most likely wouldn’t use Style Guide Guide project again
  - Gatsby Docz plugin as an alternative
  - Styling scope is an issue – style guide styles can leak into components

TO EXTERNAL OR NOT EXTERNAL

or: Where should YOU put the mustard?



YOUR NEEDS MAY BE DIFFERENT



THINGS TO CONSIDER

- Would there be value in re-using these components?
  - Sometimes hard to know for sure.
- What role do prototypes play in project?
  - Is prototyping in code important?
  - Will design system be published independently?
- How will this align with your team's workflow?
- How compatible is this with your team's makeup?
  - Will it allow more people to contribute?
- Is locking down design system desirable?

Join us for the Unconference

Saturday, June 8th, 9:30am-4:30pm

#tcdrupal

TWIN CITIES DRUPAL CAMP
tcdrupal.org
JUNE 6-8, 2019

What did you think?

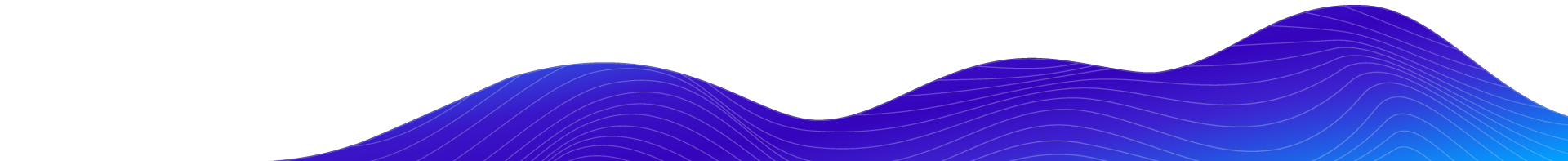
Locate this session on the TC Drupal website:

https://2019.tcdrupal.org/schedule

Leave your feedback

https://2019.tcdrupal.org/feedback

Thank You!

Brian Perry
Lead Front End Developer

Email: brian.perry@bounteous.com