UI theming with Bootstrap 4

Objective Develop a Bootstrap 4 front end theme. Define global fonts, colour palette, grids and element styles. The packaged theme must be cross platform compatible (AngularJS, Bootstrap JS, other...).

Why Bootstrap? Bootstrap offers a highly flexible and customisable SASS/CSS framework. By utilising the existing layout and styling components of the Bootstrap framework we can quickly define our own markup standards, creating fully documented working web patterns.

A custom Bootstrap theme can also accommodate an update of global styles across multiple platforms - the theme is compiled a **single CSS file** hosted on our servers.

We can minimize the need for custom CSS styling during development by including most if not all not all of the available Bootstrap components in the theme.

What is Bootstrap?

Bootstrap is a CSS and JQuery framework. We will be creating a **CSS only** Bootstrap 4 theme that will be compatible with either Bootstrap JS OR AngularJS or other platforms.
https://www.w3schools.com/whatis/whatis_bootstrap.asp

What is SASS?

SASS is a CSS preprocessor - a language that adds extra features to CSS. We will be using SASS to redefine Bootstrap default styles, creating our custom theme.

SASS must be ‘compiled’ to normal CSS before it can run in a browser.
Bootstrap 4 uses SCSS which is a variation of and the ‘next generation of SASS’.

Recommended tools for customising Bootstrap 4

Visual Studio Code
Features a Terminal window alongside code editing and file structure views.
https://code.visualstudio.com/

Gulp
Assists in a quicker workflow when working with ‘Packages’
https://gulpjs.com/

Bootstrap 4 as a local ‘Package’

In order to theme then re-compile Bootstrap 4 we must install Bootstrap 4 and all it's files locally. This can be achieved in a few ways, in this case I installed Bootstrap as a ‘Package’. Installing locally is different to the standard and simplest method of loading all of Bootstrap from their content delivery network (CDN) .

To install Bootstrap locally as a package I enabled ‘NPM’ on my local machine.
‘NPM’ or ‘Node Package Manager’ is a tool that allows the local installation of ‘packages’.
NPM commands can then be executed in Terminal (see steps below).

How to download and install NPM

https://nodejs.org/en/download/

NPM comes with NodeJS, although I won’t be using any Node JS code in this project.

After install you can check the NodeJS and NPM version by entering the following commands in Terminal:

```
node -v
npm -v
```

Installing Bootstrap 4 locally using NPM

Step 1

Create a new directory for the local site

In my case the path to the site root is:

```
~/Sites/croydon_theme_BS4/
```

Step 2

Navigate to the new directory in Terminal

Use the 'change directory' command:

```
cd ~/Sites/croydon_theme_BS4/
```

Optional step

You can check the contents of the directory you have navigated to by typing the 'print working directory' command in Terminal:

```
pwd
```

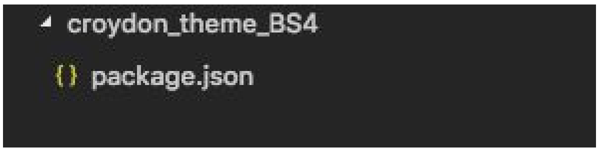
Step 3

Initiate the package

When initiating the package, a file called 'package.json' will be appear inside the site directory. This file contains key information such as the package name, description, version and more. To initiate the package, type the following command in Terminal:

```
npm init
```

Complete the steps presented in the Terminal adding the requested package information.
Hit return to skip items, these can be amended later by editing the 'package.json' file.



```
◢ croydon_theme_BS4
  {} package.json
```

Above: The contents of the directory after initiating the package

Step 4

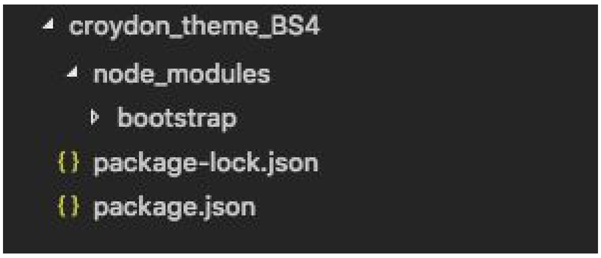
Install Bootstrap 4 with NPM

In Terminal make sure you are still browsing the desired directory and type the following command:

npm install bootstrap --save

The '--save' command is good practise as it creates a 'package-lock.json' file and makes sharing the package easier.

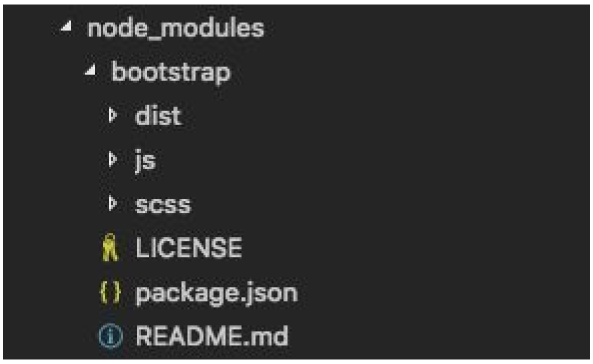
You will now see a 'node_modules' folder appear containing the Bootstrap files.



```
◢ croydon_theme_BS4
  ◢ node_modules
    ▷ bootstrap
  {} package-lock.json
  {} package.json
```

Above: The contents of the directory after installing Bootstrap with NPM

Inside the folder 'bootstrap' > 'dist' we find the compiled code.
Inside the folder 'bootstrap' > 'scss' and 'js' we find the raw pre-compiled code.
!!!!!! Do not edit anything inside any of these Bootstrap core folders as changes will be overwritten if Bootstrap is re-installed or updated.



```
◢ node_modules
  ◢ bootstrap
    ▷ dist
    ▷ js
    ▷ scss
      LICENSE
    {} package.json
    ⓘ README.md
```

Above: The contents of the 'bootstrap' directory

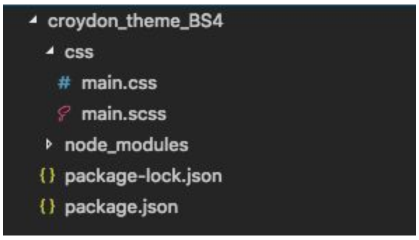
Customising Bootstrap 4

Follow steps 1- 4 before completing these steps

Step 5

Create a CSS file and SCSS file

1. In the site root create a new directory called ‘css’
2. Inside ‘css’ create a new file called ‘**main.css**’ this will hold our compiled CSS
3. Inside ‘css’ create a new file called ‘**main.scss**’ this will hold our precompiled SASS



```
◂ croydon_theme_BS4
  ◂ css
    # main.css
      main.scss
  ▸ node_modules
  {} package-lock.json
  {} package.json
```

Above: The new ‘css’ directory and SCSS files

Step 6

Install SASS compiler as a command line tool

To convert our SASS code to CSS we need a ‘SASS compiler’.

To install SASS compiler in an environment where Ruby is already running (Mac) type this command in Terminal:

```
sudo gem install sass
```

Step 7

Import required Bootstrap elements and apply custom theme

1. Use ‘**main.scss**’to set up the custom Bootstrap imports
2. Add SASS customisations to ‘**main.scss**’

Note: A custom Bootstrap Import is lighter than the full import as we can choose the required parts for example ‘grid’, 'forms’, 'navbar’ etc.

Step 8

Compiling our custom SCSS file to a CSS file

REF: http://sass-lang.com/documentation/file.SASS_REFERENCE.html

Compile the CSS file by typing this command in Terminal:

```
sass css/main.scss css/main.css
```

Note: We are working from the site root

Is a Bootstrap CSS theme compatible with all Croydon Council Web Platforms?

And which version of Bootstrap should we use?

Bootstrap 4 CSS Theme

Pros

- Better accessibility (REM units)
- Easily customised themes with SASS
- Latest Version
- Will have new features added
- Supported by Bootstrap

Considerations

- Need to investigate implication of applying to the Drupal 7 site
- Integration will be tested on My Account

Support

- No support for IE9> although fallback/fixes are available

——

Bootstrap 3 CSS Theme

Pros

- Still supported by Bootstrap

Cons

- Less accessible
- No new features will be added
- Legacy configuration, harder to customise and share
- May need to upgrade in future - requires major code re-write to update

Support

- No support for IE7> although fallback/fixes are available

——

Custom coded CSS theme

Could be considered as an alternative to Bootstrap.

Pros
- Full control over markup and accessibility

Cons
- Longer workflow everything coded from scratch

Support
- Would be coded with browser compatibility and fallbacks as far as we choose

——

Using AngularJS with the theme

Used in My Account and will be bolted into one of the above themes

AngularJS Support
- No support for IE8> - untested

——

Using Drupal 7 with the theme

Platform for the main website https://croydon.gov.uk
Doesn’t currently use Bootstrap for layout and theming.

Considerations
- Will it be upgraded to Drupal 8?
- Can we implement a Bootstrap theme?
- Which version of Bootstrap is best compatible with the Drupal Version?

(solved, theme may not be implemented on D7, D8 could be compatible)