Node.js

To start developing with this theme you're going to need to install Node.js – just download the installer (https://nodejs.org/en/download/) and install it.

The theme was programmed using two wonderful technologies – Sass (http://sass-lang.com/) for the styles and Pug (https://pugjs.org/api/getting-started.html) (FKA Jade) to generate HTML files. As well as Gulp (http://gulpjs.com/) to help manage those technologies. Gulp is also used to help optimise your resources for the production purpose.

Sass

> Sass is the most mature, stable, and powerful professional grade CSS extension language in the world.

Writing styles with Sass is a pure pleasure. We've structured Zulu's components to make them easily accessible and editable. The main file is `zulu.scss`, which contains `@import`s from all of the components, like common styles, buttons, modals, etc.

Bourbon

> A simple and lightweight mixin library for Sass.

This library provides a ton of awesome mixins, like prefixes for various properties, which really simplify cross-browser development. Make sure you have a look at Bourbon docs (http://bourbon.io/docs/) to understand how much this library really does in terms of simplifying styles building.

Variables

The color variables are defined in `components/variables.scss` file. We've also included `materialize-colors.scss` to access the `color('name', 'shade')` function, which is provided by Materialize's source and allows you to easily use the Material design palette (materializecss.com/color.html#palette).

You can also define your own media query ranges for small, middle and large displays.

Pug

This template engine runs on Node.js and allows you to be very flexible with your HTML layouts. The main file is `_base.pug` which is being extended by `index-*.pug` files and the AJAX pages are located in `pages` directory.

Package managers

npm

npm (https://www.npmjs.com/) is a famous package manager on top of Node.js. Of course, you're going to need Node.js (https://nodejs.org/en/) installed as well. Fortunately, it's relatively simple to do that, as you have to download the installer (https://nodejs.org/en/download/) and run it. We've used this to define our dev dependencies, those we need in development process. You can see them in `package.json`.

Run `npm install` to install all of the dev dependencies.

Bower

Bower (http://bower.io/) is another package manager and we've used it to define our non-dev dependencies. Those that are required for the theme to function. The dependencies are listed in `bower.json`.

Run `bower install` to install all of dependencies.

Composer

Composer (https://getcomposer.org) is used to fetch PHPMailer (https://github.com/PHPMailer/PHPMailer). Feel free to add any PHP dependecies via this package manager.

Gulp

Gulp does so many things. First, it allows us to compile our source code into usable assets. Secondly, it optimises those assets for the production. All of the commands and configurations are located in `gulpfile.js`.

Gulp installation

Gulp is installed via npm.

Following the Getting started with Gulp (https://github.com/gulpjs/gulp/blob/master/docs/getting-started.md) guide, run `npm rm --global gulp` to remove any previous versions. Then run `npm install --global gulp-cli` to install it globally or `npm install --save-dev gulp` to install it as a dev dependency of the project.

It also ensures you're not going to have any conflicts with the previous version of Gulp.

Plugins

There's a number of plugins Gulp uses to run the required tasks, you can see those on the first lines of `gulpfile.js` as well as in `package.json` under devDependencies.

Tasks

The tasks themselves are pretty self-explanatory, we've also made sure to comment the file properly. There are:

- `gulp` (default) to run `gulp sass`, `gulp pug` and `gulp watch`
- `gulp sass` to build `zulu.css` from `zulu.scss`, as well as compile `linea.css` from `linea.scss`, using source maps

- `gulp min:css` to create `zulu.min.css` and `lines.min.css`, which are optimised with cssnano and css-mqpacker. The file is placed in `assets/build/` directory. **Note**: This command depends on `gulp sass`, run it first
- `gulp pug` to build the html files. You need to specify which files you want to compile. This task also beautifies the html, making it more readable and easier to modify
- `gulp watch` that watches for the changes in your scss files, as well as pug files and runs appropriate commands (`sass` and `pug`) accordingly
- `gulp imagemin` to optimise the images found in `assets/images/` directory