

Final Project Report

Michael Taylor, and Eamonn De'Leaster*

Waterford Institute of Technology,
Dept of Computing and Mathematics,
Cork Rd, Waterford City, Ireland
`20064376@mail.wit.ie`
`http://www.wit.ie`

Abstract. A complete UI framework designed for entry level users with a focus on easy to understand terminology, introducing templating, and partials with clear, concise, customisable themes.

Keywords: CSS, User Interface, framework, beginners, templates, partials, themes

* Supervisor

Table of Contents

| | |
|-------------------------------------|----|
| Final Project Report | 1 |
| <i>M. Taylor</i> | |
| Acknowledgements | 3 |
| Introduction | 4 |
| <hr/> | |
| I user stuff | |
| <hr/> | |
| Boilerplate | 9 |
| Partials | 10 |
| Themes | 11 |
| Command Line Interface | 12 |
| <hr/> | |
| II Development | |
| <hr/> | |
| Framework | 15 |
| Conclusions | 16 |
| Future Work | 16 |
| Bibliography | 17 |

Acknowledgements

discuss framework here typography issues colour scheme font

Introduction

User Interface (UI) and Cascading Stylesheets (CSS) frameworks are abundant and readily available and generally covered by permissible licence agreements [Arsenault, 2016]. This not only allows commercial use of the frameworks but ultimately encourages tie in to a particular framework. End users expect all websites to work across their devices, laptops, tablets and phones. This requirement of responsive web applications ensures that hand coding a solution is a daunting task for entry level developers.

Using a framework allows the user to speed up the initial mock-up process, they offer clarification on common CSS issues, have wide browser support. Frameworks are good for responsive design, offer clean, and tidy code. There are disadvantages to using frameworks such as an abundance of unused code left over, this is seen more so in large front end frameworks such as Twitter Bootstrap. There is a slower learning curve with using frameworks, and the user does not learn to do it themselves through problem solving.

Frameworks such as Twitter Bootstrap [et al, 2016b], and Zurb Foundation [et al, 2016a] offer complete solutions, with ready built code for forms, buttons, fluid layouts to popovers. Skeleton [Gamache, 2016] is an example of the other end of the spectrum. Skeleton is a boilerplate for responsive, and mobile first development. It is designed to be light, and is built with less than 400 lines of code. Unlike Bootstrap or Foundation, skeleton is designed to be the users starting point, not their full solution.

This project is being built to offer the end user a complete solution designed with an entry level user in mind. The framework will be non opinionated unlike frameworks such as Bootstrap who are very opinionated about their design Guadia [2016]. This framework will be designed to be the starting point for a user, for them to add in their custom styling.

Instead of the user building large complex websites with repeating code, one of the project goals is to get the user familiar with concepts such templates, and partials. Partials break up the html code into smaller more manageable fragments that can be used across multiple html files. The framework will be built with this in mind, creating classes, and id's that can be reused instead of having a bloated package with a lot of unused code [K, 2016].

A major benefit of a project, is the ability to customise or use prebuilt themes. A prebuilt theme can allow the user to concentrate on the aesthetics more so than the structure of the project. Using a prebuilt theme does have its negatives, the theme itself can be expensive costing hundreds if not thousands. Many shortcuts could be used in the development of themes, not adhering to W3S standards. These themes can also offer little customisation [Weller, 2016].

Further attempts to incorporate Bootstrap into projects demonstrated the syntax as very unfriendly, and noticeably more difficult than hand coded CSS. Bootstrap syntax such as:

```
<div class="col-sm-4">
```

does not describe in any form that it would be displayed as a three column layout in the browser. Changing this to a one column layout by the syntax was found to be problematic and the frameworks were also found to be opinionated.

Additionally while working as a web developer mentor at Coderdojo, students were observed to have similar experiences. Many were reluctant to learn frameworks such as Bootstrap as they were too confusing.

Part I

user stuff

Boilerplate

talk about portfolio, blog and commerce website here

Partials

talk about partials here web components

Themes

Talk about themes here

Command Line Interface

talk about creating command line interface here

Part II

Development

Framework

discuss framework here typography issues colour scheme font

Conclusions

Future Work

The future work section is broken up into the following research areas:

Bibliography

- Cody Arsenault. *Top 10 Front-End Frameworks of 2016*, 2016. Available at: <https://www.keycdn.com/blog/front-end-frameworks/> [Accessed: 03/12/2016].
- Alexi Sellier et al. *Getting Started*, 2016a. Available at: <http://lesscss.org> [Accessed: 26/11/2016].
- Hampton Catlin et al. *CSS with superpowers*, 2016b. Available at: <http://SASS-lang.com> [Accessed: 26/11/2016].
- Dave Gamache. *A dead simple, responsive boilerplate.* , 2016. Available at: <http://getskeleton.com> [Accessed: 26/11/2016].
- Kemie Guadia. *CSS Frameworks- comparing Bootstrap alternatives*, 2016. Available at: <http://www.monolinea.com/css-frameworks-comparison/> [Accessed: 08/12/2016].
- Karol K. *The Bootstrap Framework Controversy ... Should You Use It or Not?*, 2016. Available at: <http://www.htmlcenter.com/blog/the-bootstrap-framework-controversy-should-you-use-it-or-not/> [Accessed: 08/12/2016].
- Nathan B. Weller. *Custom vs Pre-made WordPress Themes A Look at the Pros and Cons*, 2016. Available at: <http://wplift.com/custom-vs-pre-made-themes> [Accessed: 08/12/2016].