Library application

IDV 3

Module 1

Marileen Degenaar 130002

Alexander Skelton

24 March 2015

Tuesday 14:00 – 18:00

1 INTRODUCTION

The aim of the project was to design and develop a fully responsive website for a library to store and display their books and papers to the users. The website should react as a mobile application on small devices but remain a website on desktops.

2 LIBRARY WEB APPLICATION: THE THINKING BEHIND DEVELOPING

Creating a responsive website, I made use of the CSS framework, *Bootstrap*. The design of the application is based on the statement “Mobile first”. It looks like a cellphone application, not a website, until viewed on a larger device such as a desktop. A very clinical design style is implemented to keep the focus purely on the functionality of the application.

The mobile menu slides open from the left of the screen and the menu icon is the standard and well known “burger” icon. The menu slides open and close to save screen space on small devices like mobile phones. The menu makes use of standard bootstrap “glyphicons” / icons to increase the user experience and add to the aesthetics. These icons also add to the meaning of the links and confirm what the user think the page might contain.

The application consists of static and dynamic content. Static content is used on the “Contact us” page for the display of contact details and a Google map. Most of the application however consists of dynamic content relevant to the content in the database.

This database is a catalogue of the Library’s books and papers and I used PHP to connect to the database, insert the data into an array and encode it as JSON to create a web API. I used AngularJS to send the information, after I applied relevant filters, to the HTML document.

I chose PHP for my database connection and to create the API because personally I’m familiar with PHP and it still remains a very important language in web programming. Additionally, I used “Slim” framework to ease the creation of the API in PHP. Web APIs sends raw data which my website consumes using AngularJS.

The website is also a single page which acts as multiple pages by using panels. This creates an application that does not need to reload every page, instead just switches to the panel selected from the navigation. This was done by coding in AngularJS and making use of AJAX requests.

Dynamic content in this website is achieved by getting data from my web API and injecting them into my HTML code by using AngularJS Controllers. I created a “FilterController” which takes data from the database and searches for results matching the request from the HTML document. An example is: In the “Shop” the user has to choose from a dropdown menu according to what category he/she would like to view the books. This uses a filter to output only the books which category is the same as the category the user chose from the dropdown. Furthermore I created filters which checks if the book has recently been viewed, or is recommended for the user. These filters is then applied where necessary.

3 CONCLUSION

This application is fully functional and with the use of selected technologies, it is effectively programmed, keeping the code clean. The design layout of the application is inspired by design standards for user friendly applications and it is both aesthetically pleasing and functional. The most challenging part in the development of this application was the extensive use of custom filters in AngularJS.