**ISCG 6420 Data Internet & Website Development**

**Assignment 2**

**Group Member**

Yixin, Zhang         1483998       
 Yujia, You 1468225

Chengye, Gao 1458508

Declaration: This work has been done all by ourselves.

## Introduction

## This site---Welcome to Auckland City Books, provides an online book shop for the anyone who can surf on the internet. Its target audience is people who keep the habit of book reading. It is designed for the purpose of making the purchasing and selling flow easier and more comfortable.

## Using this site, the customers can 'visit' the book shop anytime and anywhere, as long as there is network connected. They don't have to go to the physical book shop to select books as all books and their detail information can be browsed on the webpage.

## Furthermore, it has some great features that physical book shop doesn't have: get top 5 books in terms of popularity, get booklist group by category and so on.

## The source code can also be found in GitHub

## URL:

## <https://github.com/xin080124/6421_zhangy324_Assignment1.git>

## In the subfolder: Assignment

Html:

## https://github.com/xin080124/6421\_zhangy324\_Assignment1/tree/master/Assignment

## Site map

The website is consist of 4 webpages, they are index.html, about.html, membership.html, and product.html, each of them has a navigator to anyone of the other 3 pages.

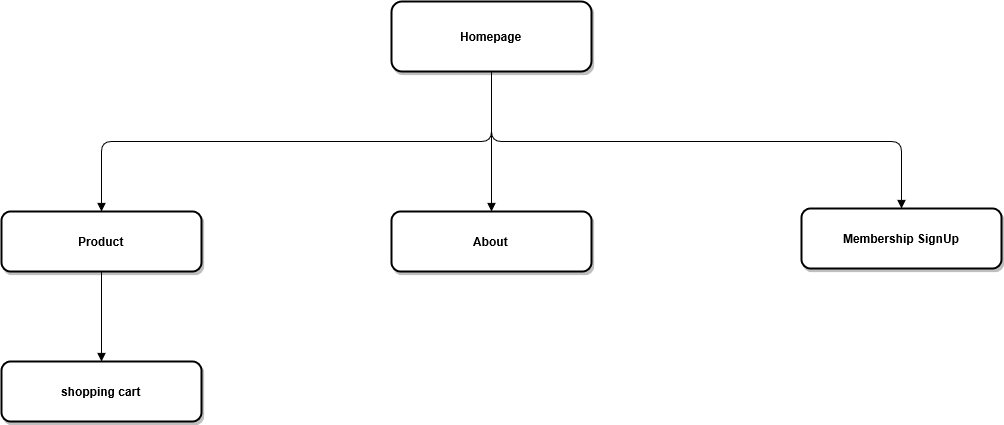


Figure1: sitemap

## Function table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pages | Function1 | Function2 | Function3 | Function3 |
| Homepage | Show top 5 popular books | Show top 5 on Sale books | Navigate to all of the other pages. |  |
| Product Page | Show 20 books and the basic information of each | A shopping cart for users to add books in and get the total price. | Show books group by category | Navigate to all of the other pages. |
| Membership Page | A form for user input to sign up membership | Validate user input and give proper feedback for user to fix the error in entered |  |  |
| About Page | Details of the online shop | A google map for user to get the exact address of the physical shop |  |  |

## Page designs for each page

Homepage:

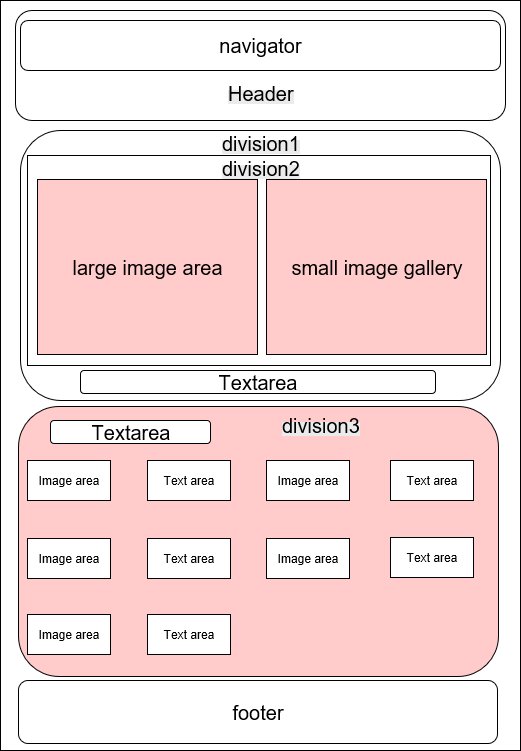


Figure2: the layout of the homepage

technical details：

The elements with pink color are dynamic, the list in these containers are generated by Ajax+javascript, while the other white color element are static.

Division 2 shows the Top 5 popular books; division 3 shows 5 on sale books, both of the sort are implemented by JavaScript.

The pink part is using dynamic elements while other parts are suing static elements.

Membership:

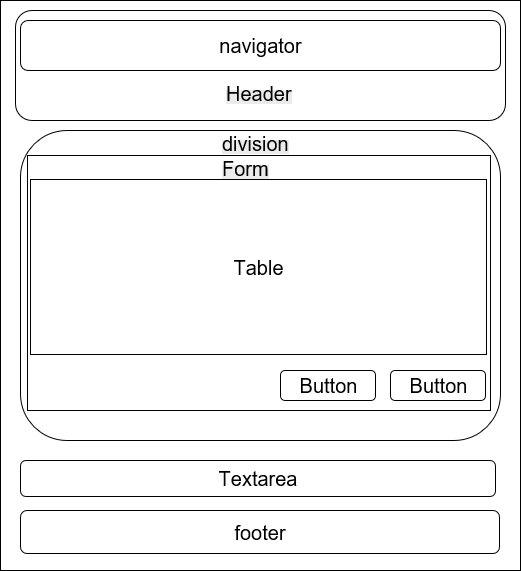


Figure4: the layout of the membership page

technical details：

All of the elements ---- textbox, check box and so on ---on this page are static, and the user input validation is implemented through JavaScript.

About

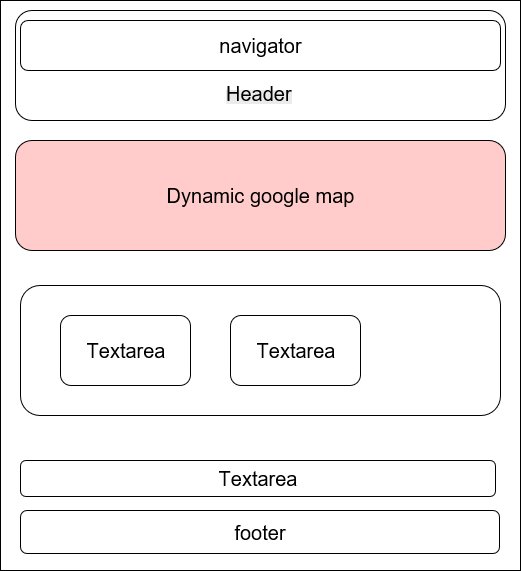


Figure5: the layout of the about page

technical details：

All of the elements ---- textbox, check box and so on ---on this page are static, and the user input validation is implemented through JavaScript.

The pink part is using dynamic elements while other parts are suing static elements.

## Where and how JavaScript/Ajax/CSS has been used on the site

In the code, JavaScript has been used in the following places:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| html file | line | file | method | aim |
| index.html | 265 | js/jquery-1.6.2.min.js  js/jquery.galleriffic.js  js/global.js js/jquery.opacityrollover.js | initArrays();  loadDoc(); | Sort books by popularity and on Sale |
| product.html | 430 | initArrays();  filterByCategory(x,category);  ShowCategory(category); | Sort books by category  Add books to shopping cart |
| membership.html | 99 | checkSubmit(); | validate user's input, if there's any error, display an appropriate error message |
| About.html |  |  |  |

Ajax technique is used to read the xml file:

It is implemented by the following lines:

|  |
| --- |
| var xhttp = new XMLHttpRequest();  xhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) {  myFunction(this);  }  };  xhttp.open("GET","books1652.xml?t=" + Math.random(),true);  xhttp.send(); |

Here we create a XMLHttpRequest first and then used to generate and send a “GET” request to the server， this process is asynchronous so it would not block the browser. The param “"books1652.xml?t=" + Math.random()” is to generate a different url with every request to make sure the browser read data from the file on remote server, not the data from browser cache. After fetching the data, we use some JavaScript method to sort in terms of some attribute, then show the top 5 popular list and top 5 on sale list on the Home page; we also use some JavaScript method to sort in terms and show different types of books in product page.

We use a series of external CSS files to enable all the pages share the same header, footer, body font when it is necessary, and also, have their own design in other layout. It’s implemented through codes like

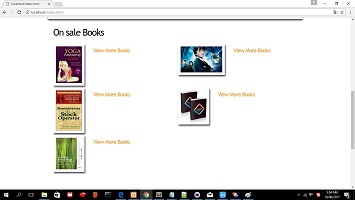
|  |
| --- |
| <link rel="stylesheet" href="css/reset.css" type="text/css" media="screen">  <link rel="stylesheet" href="css/style.css" type="text/css" media="screen">  <link rel="stylesheet" href="css/grid.css" type="text/css" media="screen"> |

in each page.

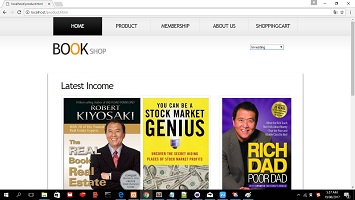
## Evidence of testing

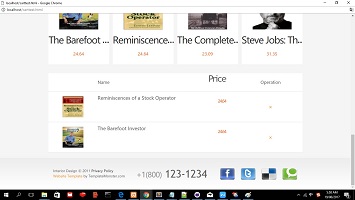
Our code works ok on chorm, the following are some screen shot:

Sort function on Homepage:



Category function on product page:



Shopping cart on shopping cart page:

## Future improvement

1. We can separate the shopping cart into another page.

2. Customers should have an account to pay the bill online or save the deliver address.

3. The books can show some comments which provide by previous customers.

4. Product page should allow customers browse books by different ways such as by publisher.

5. Customers should able to give a mark to the book. The results will show on the product page.

6. We maybe need a chat window so the customers can chat with our service.