**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 a 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>

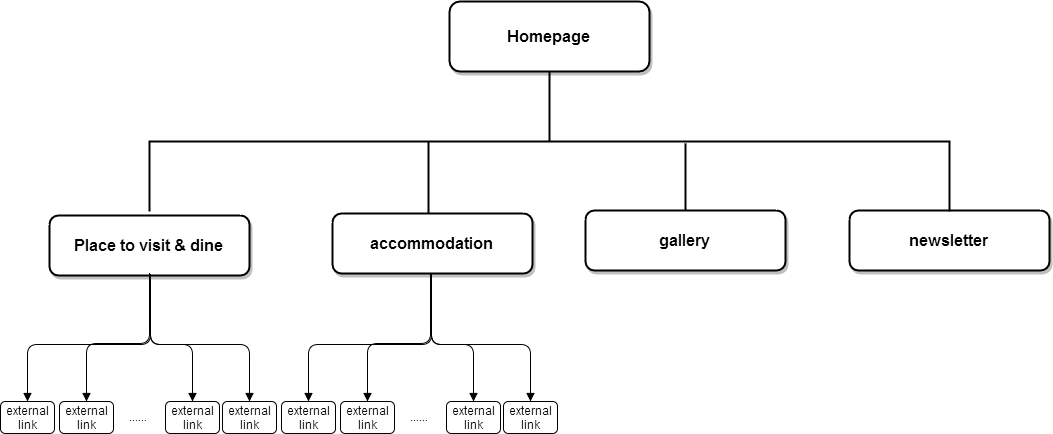
## In the subfolder: Assignment

Html:

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

## Site map

The website is consist of 5 webpages, they are home page, product page, membership signup page, and about page.html, each of them has a navigator to anyone of the other 4 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:

indexStructure

Figure2: the layout of the homepage

technical details：

Header：use <img src="images/qslong.jpg" class="img-responsive" /> to show image;

Navigator：the elements are put in <nav> <ul class="nav"> <li class="active">，as in the CSS file, the .nav li class is defined as:

.nav li

{

position:relative;

display:inline-block;

zoom:1;

\*display:inline;

}

The display attribute value is inline-block,so the elements in the navigator can show in a line.

Media frame：use <iframe width="560" height="315" src="https://www.youtube.com/embed/rGKf1NhtJ1o"

frameborder="0" allowfullscreen></iframe> to show video;

EventContainer: events basic information and external link are put in <div class="main-container"><div class="box">, so that the CSS style div.main-container

{display:block;text-align:left;margin:0 0 1px} and .box{...} would apply, the .main-container style defined text-align:left; so that the texts of events show in form of left-align; the .box style has some attribute like border and -webkit-border-radius so it can generate a solid line rectangle on the webpage;

## Page designs for each page

Event external links are imported by codes like :<a href="https://www.queenstownnz.co.nz/things-to-do/events/

">see more events &raquo;</a>

Footer: footer has been divided into 2 parts, left part and right part.

The left part is in <p class="tagline\_left"> tag, and the right part is in <p class="tagline\_right"> tag.

in CSS file, the .tagline\_left and .tagline\_right are defined as:

.tagline\_left{float:left} , .tagline\_right{float:right}, and their opposite float attribute enable the 2 text block appear at left and right respectively.

It should be noted that the explanations of the navigator and footer layout also apply to the other 4 html files(visitAndDine.html, Accommodation.html，Gallery.html and Newsletter.html ) as they share the same navigator and footer.

visitAndDine

dineStrcture

Figure3: the layout of the visitAndDine page

technical details：

The left block lists a series of places to visit and the activities there; while the right block lists the dining places, each sub block on the page has a brief introduction of one place and the external link to the official website.

Both the left and the right block are put in the <div class="main-container">, <div class="box">,but in the <div class="box">, the left and right block are put in different sub div, the left block is put in <div class="block">, in its css is like this:

#home\_featured21 .block{display:inline;float:left;width:510px;margin:0 30px 0 0;padding:0}

The “float” value is left,so this block appears at the left side of the page;

The right block is put in <div class="block last">,in css, it is like this:

#home\_featured21 .last{float:right；width:275px;margin-right:0}

he “float” value is right,so this block appears at the right side of the page;

Images are showed by code like <img src="images/skydive.png" width = 500px alt="" />;

Text are put in <p></p>

And external link are implemented by code like:

<p class="readmore"><a href="http://www.nzoneskydive.co.nz/tandem-skydive/skydiving-in-queenstown?language=en-nz

">View More Information &raquo;</a></p>

## Page designs for each page

Accommodation

accommodationLayout

Figure4: the layout of the accommodation page

technical details：

This page list the accommodation that the tourist can choose, each accommodation is provided with an image, basic information and external link. All of these contents are put in <div class="main-container">, <div class="box">, <div class="portfolio">and for each accommodation, the basic information and external link is listed in <div class="tagline\_left"> and the image is listed in <div class="tagline\_right">, in CSS, the classes are defined as

.portfolio .tagline\_left

{

display:block;

float:left;

width:500px;

line-height:1.6em;

margin:10px 0 0

}

## Page designs for each page

.portfolio .tagline\_right

{

display:block; float:right; width:600px

}

so the fist part is at the left side while the second are listed at the right side.

Images are put by <img src="images/HiltonHotel.jpg" alt="" />

As in CSS file , there are styles defined as :

.portfolio .tagline\_right img

{display:block;width:470px;height:auto;border:10px solid #E0E0E0;padding:5px}

.portfolio .tagline\_right img:hover

{display:block;width:470px;height:auto;border:10px solid #B4B4B4;padding:5px}

So images on this page all have a border and their effects will change when mouse hovers on.

The text paragraphs are put in <p></p> and the external link are put in codes like <p class="readmore"><a href="http://www3.hilton.com/en/hotels/new-zealand/hilton-queenstown-resort-and-spa-

ZQNHQHI/index.html">View More Information &raquo;</a></p>

Button Group：there are five buttons in the button group, each has a category name. When user clicks anyone of them, only the accommodation of the clicked category will show while the other categories will hide.

The button widgets on the html file are defined like this:

<li><input type="button" id="butn1" value="All categories" onclick="toggle('all')" /></li>

<li><input type="button" id="butn2" value="hotels" onclick="toggle('hotels')" /></li>

<li> <input type="button" id="butn3" value="motels" onclick="toggle('motels')" /> </li>

<li>

<input type="button" id="butn4" value="lodges" onclick="toggle('lodges')" />

</li>

<li class="last">

<input type="button" id="butn5" value="Bed & Breakfast" onclick="toggle('bb' )

So buttons have different parameters for toggle() when onclick event is triggered.

And then in js file, using the parameter, the function toggle() would generate different hiding items list and showing items list, thus a accommodation filter is implemented.

## Page designs for each page

Gallery

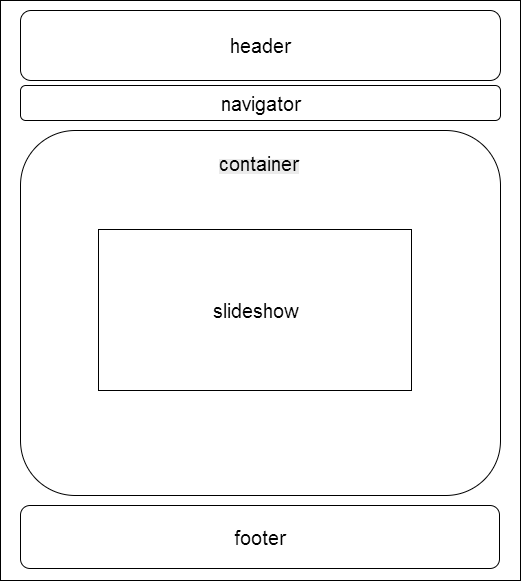


Figure5: the layout of the gallery page

In this page, all the slideshow images are put in <div class="fixbox">,<div class="ppt-container">and <ul class="image-list"> ,and I use a third-party code---jQuery code to control the images show/hide. Each image has a unique index, so in javaScript, I use

$(".ppt-container .image-list li[data-index="+ iIndex +"]").fadeIn(1500); to show the image-list li[iIndex];

And use $(".ppt-container .image-list li[data-index="+ iPreIndex +"]").fadeOut(1500); to hide the image previous showed.

There is also a button list at the bottom of the images, user can click one of them to see a certain image. When the user click the iIndex button, like written above, the number iIndex image would show while other images would hide.So this is manual play; I also implemented auto play by simulating a user click event every 5 seconds in js code:

setInterval(

function() //parameter 1 of setInterval

{

var iNextIndex = (iPreIndex + 1) % iCountOfImage;

$(".ppt-container ul.button-list li span[data-index="+ iNextIndex +"]").click();

},

5000

);

So this page supports both manual slide show and auto slide show.

## Page designs for each page

Newsletter

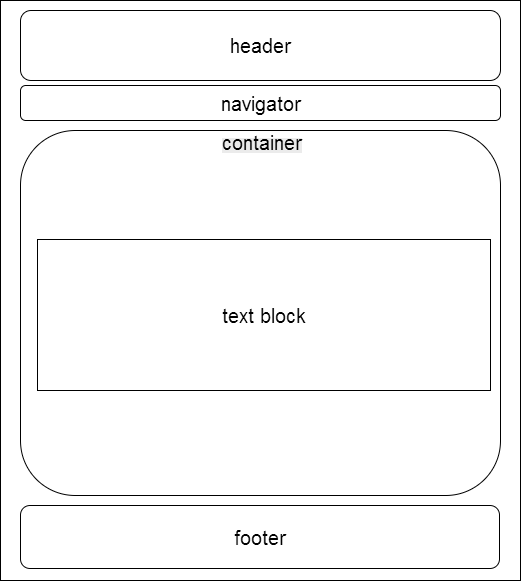


Figure6: the layout of the newsletter page

technical details:

There is no image or external link on this page, but only a form contains 3 textboxes, a checkbox and a submit button. These widgets are put in the <div class="main-container"><div class="box">, and each widget has a unique ID so that their value can be retrieved by javascript code like:document.getElementById("surname").value;

In js/global.js file, through several simple functions, these widget values are firstly retrieved and then be used to verify if they are correct. If not, the function checkSubmit() would show a message box to remind the user, by the alert() method.

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

In the code, JavaScript has been used in the following places, and all methods are put in js files:js/global.js:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| html file | line | file | method | aim |
| index.html | 265 | js/global.js | initArrays();  loadDoc(); | Sort books by popularity and on Sale |
| product.html |  |  |  |  |
| membership.html | 99 | js/global.js | checkSubmit() | validate user's input, if there's any error, display an appropriate error message |

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 ; xxxxxxxxxshopping cart

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.

## Where and how CSS has been used on the site.

The site has several html files, although they are separated, when it comes to some

sections, they share some styles, so these styles are put in one css file:"css/global.css"

so that it can be referenced by all the html files through

<link rel="stylesheet" type="text/css" href="css/global.css">

1 header

On all of the html files, the header is made by

<div class="main-container">

<header>...

And the style is defined as:

div.main-container{...}

2 Navigation bar

On all of the html files, the navigation bar is made by

<ul class="nav"> and <li class="active">...

and the style in <ul> and <ul li> are defined as:

...

.nav li {...},.nav li

.active li a{...}

...

3 in container

In accommodation.html and visitAndDine.html, the items are put in boxes, and the box is made

by <div class="main-container">, <div class="box">

Which are difined as

div.main-container{...}(also used in header), .box{...}

Moreover, the css file allows all the pages share the same defalt font, text color and so on.

By code like this:

body{...}

## Summary

Through designing and implementing this simple website of promoting Queenstown, I have understood better about web development knowledge gained so far in the course, as well as improved my skills on using JavaScript and CSS. There must be some mistakes in the work, it is not perfect. However, it is definitely a base for further study.

A list of future improvements:

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.

Reference