

# ***BOOK LAKE***

AN ONLINE READING E-BOOK

*S3677615 Shuo Wang & s3736719 Ming-jin Yeh*

*Cloud Computing 2019*

## **Summary**

Our goal for this assignment is to apply the cloud computing concept and design a highly scalable application. From this point, we have utilized Google App Engine, Google cloud storage, Google book API, Google virtualization API and AWS simple storage. The following report details the work we have done. Our application provides a user-friendly interface to access the online books in google book database. The search function returns the accurate results with the basic information to appease the users. With the help of APIs, the product provides an online reading platform for book lovers. We also devote to acquire data from various sources. It gave us a chance to exam and excel those cutting-edge products from the field. Authentication is the other feature that we have implemented in our product. In all, the web application has the potential to become a real product with more research on UI and UE design.

## **Introduction**

Recently, there are many Online Library which provides service about renting books online, or providing e-books, such as Open Library. However, the dataset in this website cannot catch up with popular search and slow. For example, if 'Python' is input to the input field, it will generate books which begin from 1979 or even show a real animal book. However, it is a very popular computer language in recent years, which is studied by lots of IT practitioner.

What is more, many books are needed to be borrowed, which is unable to meet the ever-changing technological developments, compared with electronic edition, and it is not conducive to rapid spread of knowledge. Thus, we got an idea about generating a website which can provide huge amount of popular free books for readers and all of the resource is based on the internet and sharing knowledge to the vast number of intellectual seekers.

With the development of Internet and electronic products, more and more people would like to read books by mobile phone, iPad or monitor because it is portable, easy to be obtained and save for a long time. In the sharing economy environment, people decrease the cost for getting books, such as free to watch books with advertisements or a VIP without advertisement.

This application provides a website which can be used to retrieve the information of books based on the title of the book according to the google books API. If users are interested in this book, they can read this book online to access full pages of book. Google books API will return popular books, which may save time for users. What is more, this website also provides an analysis and present it to a pie chart based on the dataset.

However, there is a few drawbacks in our designed website. Because it owns a huge amount of values stored in database, it is hard to catch dataset and save into a NOSQL, which is hard to collect information from users and for further processing. For example, which book is more popular in this month? Which book will get a higher rating? It is hard to implementation recommendation system for users who some people like the same books, they may have similar interesting in other books and so forth.

## **Related work**

The website for Open Library provides some similar functions as online book browser. In the site, they provide login and sign up function and an input box which can retrieve a book information based on the name of the book. We also gain a lot help from Google books documentation and the tutorials videos on YouTube.

## **Software design/architecture**

### **1. Google books API v1(Experimental):**

This API can perform full-text searches and retrieve book information with the following methods described in the Google Books Developer Guide [1].

- Search and browse through the list of books that match a given query.
- View information about a book, including metadata, availability and price, links to the preview page.
- Manage your own bookshelves.

### **2. Google embedded viewer API**

This API can allow book contents which come from Google books directly in the web pages with JavaScript. The Developer's Guide [2] explains how the API can be loaded using the Google AJAX loader and drawn into a particular container on your page. After embedding successfully, it can show the book details like a pdf software viewer which can get the current page number and navigates.

### **3. Google App Engine**

The website is presented in the google app engine because developer and user can get lots of benefits from it. Google App Engine[3] is a fully managed, serverless platform for developing and hosting web applications at scale and different popular computer languages, libraries, and frameworks can be selected to develop these apps, which means it can take care of provisioning servers and automatically scaling app instances based on demand.

### **4. Google visualization API [4]**

It can present data values to a chart which is easier to see the difference and the pattern.

### **5. Google Cloud Storage**

This storage can be available in the world-wide storage and retrieval of any amount of data. In our applications, this one is used to store users' information and a dataset which is used to be analysed by generating a pie chart.

### **6. S3 Bucket**

Store an image which is presented in the website.

# Implementation

## 1. Main page website

The purpose of this page is to set a primary interface for users who has registered and login to the system, and the function of this page is used to obtain users' query information about book titles that will be used to search relatively information from google books API v1(Experimental).

Type= 'text' will collect what users input information after users clicking the button.

```
<input id = "search-box" type = "text" class = "form-control" placeholder="Please Search Books!">
```

```
<button id = "search" class="btn btn-primary" onclick="">Search</button>
```

And then import the libraries, such as ajax, bootstrap.

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
```

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
```

## 2. App.js

This file will be used to process the action for search button and get results from book API in JSON file level and then decompose it to each part which what we want, such as title, author, publisher, book link, book ISBN, book image. After that, all the results will present in htmlCard and as an output for each book. After that, all the books will be present in the web page.

placeHldr will present that some of the results do not have an image.

```
placeHldr = '';
```

After user click the search button, if the input box is not null, the value in the input box (searchData) will be linked to another url (bookUrl) which is used to search the information of this book.

```
$.ajax({
    url: bookUrl + searchData,
    dataType: "json",
    success: function(response){
```

```
console.log(response)
```

In this part, it is clear to see that, after user input python, it will return 10 arrays which include many different information inside. If you cannot see content like this, there should be a some problem in returning values.

```
// if there are no results for this book, it will reminder user re-enter again

if(response.totalItem===0){

    alert("there is no results, please change a name and research again")

}else {

// valide return values, fix the format and put into the displayResults function

$("#title").animate({'margin-top': '5px'}, 1000);

$(".book-list").css("visibility", "visible");

displayResults(response); }},

error: function(){

    alert("got sth wrong");}));
```

Notice that: The response is what we got the value from google book API in JSON datatype and the console.log(response) will help developer to check does results reflect correct or not for debugging.

```
function displayResults(res){

for(var i=0; i< res.items.length; i+=2){

    item = res.items[i];

    title1 = item.volumeInfo.title;

    author1 = item.volumeInfo.author;

    publisher1 = item.volumeInfo.publisher;

    bookLink1 = item.volumeInfo.previewLink;

// there are two identifiers, 10 and 13 isbn values

bookIsbn1 = item.volumeInfo.industryIdentifiers[0].identifier;

// get image from volumeInfo.imageLinks
```

```
bookImg1 = (item.volumeInfo.imageLinks) ? item.volumeInfo.imageLinks.thumbnail :  
placeHldr;
```

formatOutput function will help format the value type to specific type in above and the parameter is bookImg, title, author, publisher, bookLink and bookIsbn.

```
formatOutput(bookImg1, title1, author1, publisher1, bookLink1, bookIsbn1)
```

```
// handling error displaying empty search box
```

```
function displayError(){
```

```
    alert("search should not be empty");}
```

### 3. Book.html

For embedded viewer API, please check the documentation in the “Hello, World” of the Embedded Viewer API.

Viewer.load('ISBN:'+bookIsbn); will be loaded the book content in the website page.

### 4. Index.php

This file is used to generate a pie chart use google visualization API. All the database file is got from google data storage, bucket. After get this file, separate each data value, and then use it to create pie chart.(please check the google visualization API document: <https://developers.google.com/chart/interactive/docs/gallery/piechart>)

## User Manual

1. You will start with the login page.
2. Login or register as a user.
3. Once login, you will see the main page.
4. In search bar, type in whatever you want to search.
5. The page will show the result as cards.
6. Click on the read button, which will direct you to the online reading browser.
7. The chart button will show you the genres percentage in the book database.
8. You can always sign out and sign in at any time.

## Reference

[1] Books APIs. 2019. Google. [ONLINE] Available at: <https://developers.google.com/books/docs/overview>. [Accessed 18 October 2019].

[2] Developer's Guide. 2019. google. [ONLINE] Available at: [https://developers.google.com/books/docs/viewer/developers\\_guide](https://developers.google.com/books/docs/viewer/developers_guide). [Accessed 18 October 2019].

[3] Google cloud. 2019. Google App Engine Documentation. [ONLINE] Available at: <https://cloud.google.com/appengine/docs/>. [Accessed 18 October 2019].

[4] Google Charts. 2019. Visualization: Pie Chart. [ONLINE] Available at: <https://developers.google.com/chart/interactive/docs/gallery/piechart>. [Accessed 17 October 2019].