Text Search Engine

--Getting Book Information

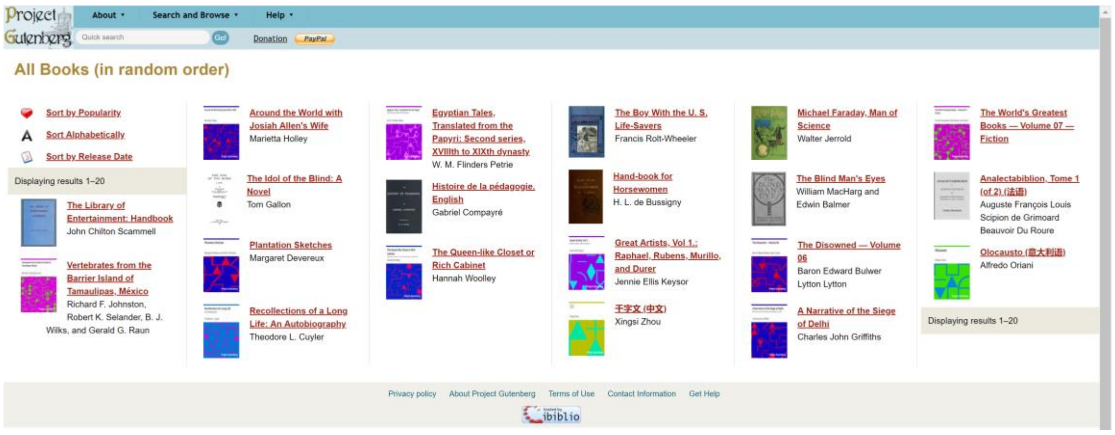
Evelyn	1830025062
Faye	1830026013
Connor	1830026020

Contents

Text files collection

Basic information

Due to our target is a text search engine, I select a free e-book database to get our data. Project Gutenberg offers over 60,000 e-books and is very easy to crawl. The website provides a random recommendation function, which can provide 20 different books each time when the page is refreshed. *(http://www.gutenberg.org/ebooks/search/?sort_order=random)* Using this page for crawling, I can control the amount of data crawled very well. For each book, I save it as a .txt file named after the book name. In order to test our project more comprehensively, I downloaded a total of 3 datasets, First a test dataset, 100 books/80MB. Second formal dataset, 700 books/300MB. Third a large capacity dataset, 7 books/2G dataset. In addition, I also crawled book titles, book authors, web links, and other additional information as auxiliary information.



All Books (in random order)

Sort by Popularity
Sort Alphabetically
Sort by Release Date

Displaying results 1–20

The Library of Entertainment: Handbook
John Chilton Scammell

Vertebrates from the Barrier Island of Tamaulipas, México
Richard F. Johnston, Robert K. Selander, B. J. Wilks, and Gerald G. Raun

Around the World with Josiah Allen's Wife
Marietta Holley

The Idol of the Blind: A Novel
Tom Gallon

Plantation Sketches
Margaret Devereux

Recollections of a Long Life: An Autobiography
Theodore L. Cuyler

Egyptian Tales, Translated from the Papyri: Second series, XVIIIth to XIXth dynasty
W. M. Flinders Petrie

Histoire de la pédagogie. English
Gabriel Compayré

The Queen-like Closet or Rich Cabinet
Hannah Woolley

The Boy With the U. S. Life-Savers
Francis Rolt-Wheeler

Hand-book for Horsewomen
H. L. de Bussigny

Great Artists, Vol 1.: Raphael, Rubens, Murillo, and Durer
Jennie Ellis Keysor

千字文 (中文)
Xingsi Zhou

Michael Faraday, Man of Science
Walter Jerrold

The Blind Man's Eyes
William MacHarg and Edwin Balmer

The Disowned — Volume 06
Baron Edward Bulwer Lytton Lytton

A Narrative of the Siege of Delhi
Charles John Griffiths

The World's Greatest Books — Volume 07 — Fiction

Analectabiblion, Tome 1 (of 2) (法语)
Auguste François Louis Scipion de Grimoard Beauvoir Du Roure

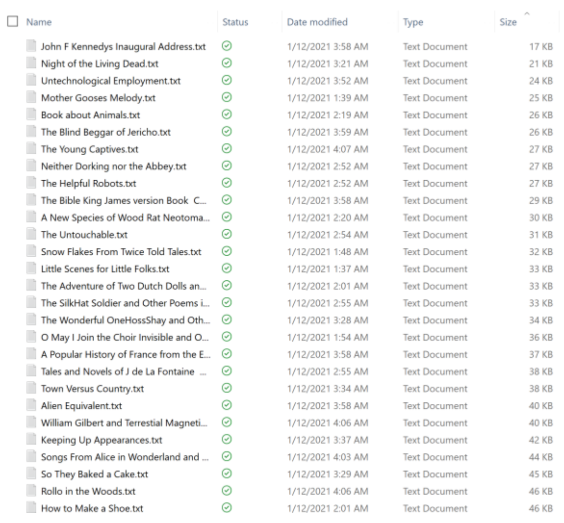
Olocausto (意大利语)
Alfredo Oriani

Displaying results 1–20

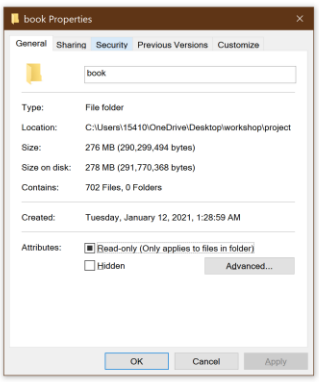
Privacy policy About Project Gutenberg Terms of Use Contact Information Get Help

Analysis of results and improvement

Project Gutenberg stores all kinds of books from all over the world, apart form English books we want, there are also French, Italian, Greek and so on, but this causes a problem. Some Italian, French, Greek letters cannot be used in the file name. And some special characters such as @#￥ are also invalid in a filename. Here is my solution: For special symbol, I define a Punctuation set to remove them if they appear in book name. For Non-English characters, I use isalpha function to test the book name and remove those books.



Name	Status	Date modified	Type	Size
John F Kennedys Inaugural Address.txt	✓	1/12/2021 3:58 AM	Text Document	17 KB
Night of the Living Dead.txt	✓	1/12/2021 3:21 AM	Text Document	21 KB
Untechnological Employment.txt	✓	1/12/2021 3:52 AM	Text Document	24 KB
Mother Gooses Melody.txt	✓	1/12/2021 1:39 AM	Text Document	25 KB
Book about Animals.txt	✓	1/12/2021 2:19 AM	Text Document	26 KB
The Blind Beggar of Jericho.txt	✓	1/12/2021 3:59 AM	Text Document	26 KB
The Young Captives.txt	✓	1/12/2021 4:07 AM	Text Document	27 KB
Neither Dorking nor the Abbey.txt	✓	1/12/2021 2:52 AM	Text Document	27 KB
The Helpful Robots.txt	✓	1/12/2021 2:52 AM	Text Document	27 KB
The Bible King James version Book C...	✓	1/12/2021 3:58 AM	Text Document	29 KB
A New Species of Wood Rat Neotoma...	✓	1/12/2021 2:20 AM	Text Document	30 KB
The Untouchable.txt	✓	1/12/2021 2:54 AM	Text Document	31 KB
Snow Flakes From Twice Told Tales.txt	✓	1/12/2021 1:48 AM	Text Document	32 KB
Little Scenes for Little Folks.txt	✓	1/12/2021 1:37 AM	Text Document	33 KB
The Adventure of Two Dutch Dolls an...	✓	1/12/2021 2:01 AM	Text Document	33 KB
The SilkHat Soldier and Other Poems i...	✓	1/12/2021 2:55 AM	Text Document	33 KB
The Wonderful OneHossShay and Oth...	✓	1/12/2021 3:28 AM	Text Document	34 KB
O May I Join the Choir Invisible and O...	✓	1/12/2021 1:54 AM	Text Document	36 KB
A Popular History of France from the E...	✓	1/12/2021 3:58 AM	Text Document	37 KB
Tales and Novels of J de La Fontaine ...	✓	1/12/2021 2:55 AM	Text Document	38 KB
Town Versus Country.txt	✓	1/12/2021 3:34 AM	Text Document	38 KB
Alien Equivalent.txt	✓	1/12/2021 3:58 AM	Text Document	40 KB
William Gilbert and Terrestial Magneti...	✓	1/12/2021 4:06 AM	Text Document	40 KB
Keeping Up Appearances.txt	✓	1/12/2021 3:37 AM	Text Document	42 KB
Songs From Alice in Wonderland and ...	✓	1/12/2021 4:03 AM	Text Document	44 KB
So They Baked a Cake.txt	✓	1/12/2021 3:29 AM	Text Document	45 KB
Rollo in the Woods.txt	✓	1/12/2021 4:06 AM	Text Document	46 KB
How to Make a Shoe.txt	✓	1/12/2021 2:01 AM	Text Document	46 KB



book Properties

General | Sharing | Security | Previous Versions | Customize

book

Type:	File folder
Location:	C:\Users\15410\OneDrive\Desktop\workshop\project
Size:	276 MB (290,299,494 bytes)
Size on disk:	278 MB (291,770,368 bytes)
Contains:	702 Files, 0 Folders
Created:	Tuesday, January 12, 2021, 1:28:59 AM
Attributes:	■ Read-only (Only applies to files in folder)
	☐ Hidden

Indexing

In this part, we will go through the text file and create an index file. This file will contain the words we capture from all documents and file name which the word locates in.

The use of Map Reduce

We had collected amount of textual data about novels. In order to make indexing efficiently, we use MapReduce to complete the process of doing it. The algorithm has three parts which are Map, Combine, and Reducing. The word frequency statistics and document list generation cannot be completed simultaneously through a Reduce process, so a Combine process must be added to complete the word frequency statistics.

Step 1: Map

Do an analysis of the text data that crawled down from the novel. Analysis the input key (word) and value (word frequency and file name)

Step 2: Combine

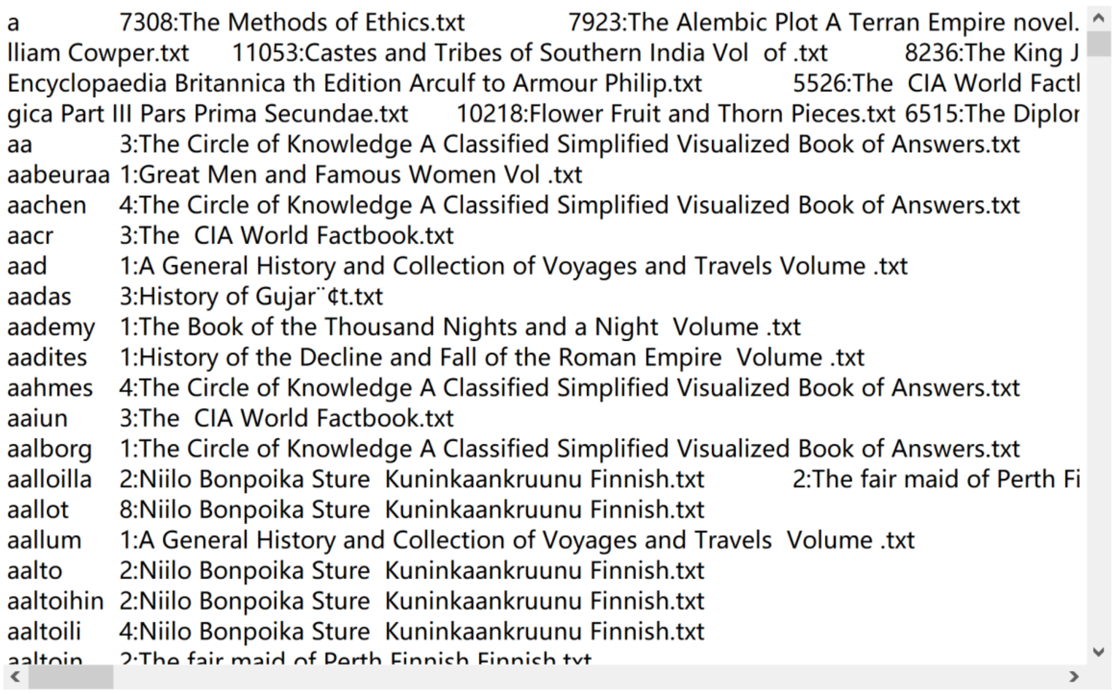
Add the values with the same key value together and get the word frequency of a word in the documents.

Step 3: Reduce

Combine values with the same key value into an inverted index file

The result of indexing

The indexing algorithm will output the words it scans from all documents. Then, it will show the text name of books which the word in it and the frequency of the word in the book in pairs. The result of indexing is as bellow.



```
a           7308:The Methods of Ethics.txt            7923:The Alembic Plot A Terran Empire novel.
lliam Cowper.txt      11053:Castes and Tribes of Southern India Vol  of .txt          8236:The King J
Encyclopaedia Britannica th Edition Arculf to Armour Philip.txt           5526:The  CIA World Factl
gica Part III Pars Prima Secundae.txt       10218:Flower Fruit and Thorn Pieces.txt 6515:The Diplor
aa          3:The Circle of Knowledge A Classified Simplified Visualized Book of Answers.txt
aabeuraa 1:Great Men and Famous Women Vol .txt
aachen    4:The Circle of Knowledge A Classified Simplified Visualized Book of Answers.txt
aacr        3:The  CIA World Factbook.txt
aad         1:A General History and Collection of Voyages and Travels Volume .txt
aadas       3:History of Gujar¨¢t.txt
aademy    1:The Book of the Thousand Nights and a Night  Volume .txt
aadites    1:History of the Decline and Fall of the Roman Empire  Volume .txt
aahmes    4:The Circle of Knowledge A Classified Simplified Visualized Book of Answers.txt
aaiun       3:The  CIA World Factbook.txt
aalborg    1:The Circle of Knowledge A Classified Simplified Visualized Book of Answers.txt
aalloilla  2:Niilo Bonpoika Sture  Kuninkaankruunu Finnish.txt          2:The fair maid of Perth Fi
aallot      8:Niilo Bonpoika Sture  Kuninkaankruunu Finnish.txt
aallum     1:A General History and Collection of Voyages and Travels  Volume .txt
aalto       2:Niilo Bonpoika Sture  Kuninkaankruunu Finnish.txt
aaltoihin  2:Niilo Bonpoika Sture  Kuninkaankruunu Finnish.txt
aaltoili    4:Niilo Bonpoika Sture  Kuninkaankruunu Finnish.txt
aaltoin     2:The fair maid of Perth Finnish Finnish.txt
```

Ranking

In order to implement a search engine query, we use the output file from indexing to do the ranking. It will decide the position at which a particular file name appears after we do the search. For example, if we search “great” in our search engine, we can get all the file name contains the word “great” which is in a proper order.

TF-IDF model for ranking

1.The introduction to TF-IDF

1) **TF (Term Frequency)**

   TF represents the correlation between a term and a document. It calculates the frequency of a term in a document.

$$TF_w = \frac{\text{the number of occurrences of a word } w \text{ in a class}}{\text{the number of words in the class}}$$

   It can be represented by

$$tf_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

2) **IDF (Inverse Document Frequency)**

It represents the weight of the word in the document. It is mainly compared by the number of documents containing the word and the total number of document sets. The more times it occurs, the smaller the weight.

$$\mathrm{IDF} = \log\left(\frac{\text{the total number of documents in the corpus}}{\text{the number of documents containing the word } w + 1}\right)$$

3) **TF-IDF (term frequency-inverse document frequency)**
The TF-IDF model can extract the most informative words. This model can assess the importance of a word to a document in the document set or corpus. The formula to calculate TF-IDF is just TF multiply IDF.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i + 1}\right)$$

It is obviously that the greater the TF-IDF of a certain word in the article, the higher importance of the word in it. Thus, we calculate the TF-IDF value for each word in the document,

2.TF-IDF ranking algorithm

This algorithm also uses MapReduce computing framework to implement it. This program has three jobs, which are represented by Job 1, Job 2, and Job 3. We first calculate TF for each word, and then calculate IDF. Finally, use the formula to calculate the TF-IDF value for each word in the document, and then arrange them in descending order, taking the first few words. When we search by word, we can get the highest articles of the word TFIDF (in descending order according to the size of TFIDF). This means that we can get a few articles that are best represented by the word.

1) **Job 1**
Input: All English text about books.
Output: The value of TF for each words in the documents.
2) **Job 2**
Input: The file outputted by Job 1.
Output: The value of IDF for words.
3) **Job 3**
Input: The file outputted by Job 1 and the file outputted by Job 2.
Output: The value of TF-IDF for words.

Ranking in python

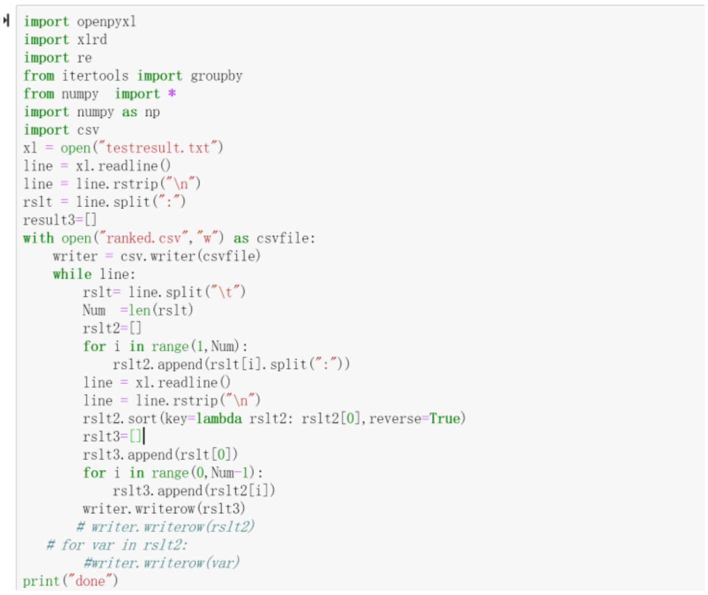
It is the algorithm we use to do the ranking without the use of TF-IDF model. The step of doing ranking in this algorithm is as below.

1) Read the file which is already complete Indexing.

2) Split the word and the list which includes the frequency and the book name in pairs.

3) Split the frequency of the word in the book and the book name.

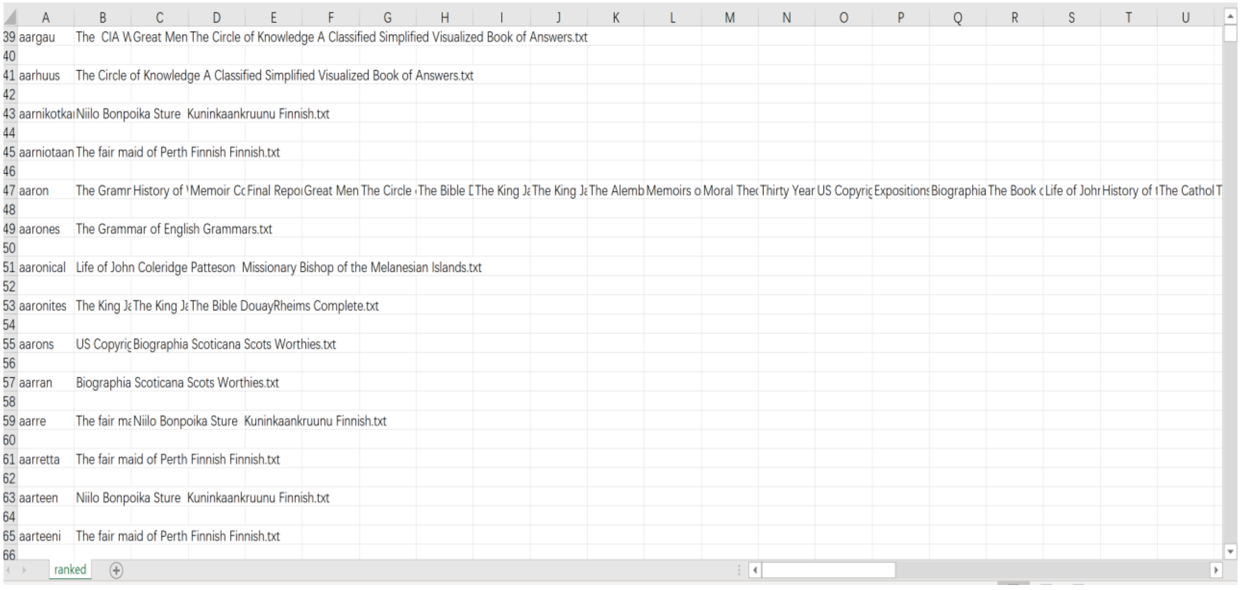
4) Capture the frequency and use it to sort the book in the decreasing order. Finally, it will return the list of the book name.



```
import openpyxl
import xlrd
import re
from itertools import groupby
from numpy  import *
import numpy as np
import csv
xl = open("testresult.txt")
line = xl.readline()
line = line.rstrip("\n")
rslt = line.split(":")
result3=[]
with open("ranked.csv","w") as csvfile:
    writer = csv.writer(csvfile)
    while line:
        rslt= line.split("\t")
        Num  =len(rslt)
        rslt2=[]
        for i in range(1,Num):
            rslt2.append(rslt[i].split(":"))
        line = xl.readline()
        line = line.rstrip("\n")
        rslt2.sort(key=lambda rslt2: rslt2[0],reverse=True)
        rslt3=[]
        rslt3.append(rslt[0])
        for i in range(0,Num-1):
            rslt3.append(rslt2[i])
        writer.writerow(rslt3)
       # writer.writerow(rslt2)
   # for var in rslt2:
        #writer.writerow(var)
print("done")
```

The result of ranking

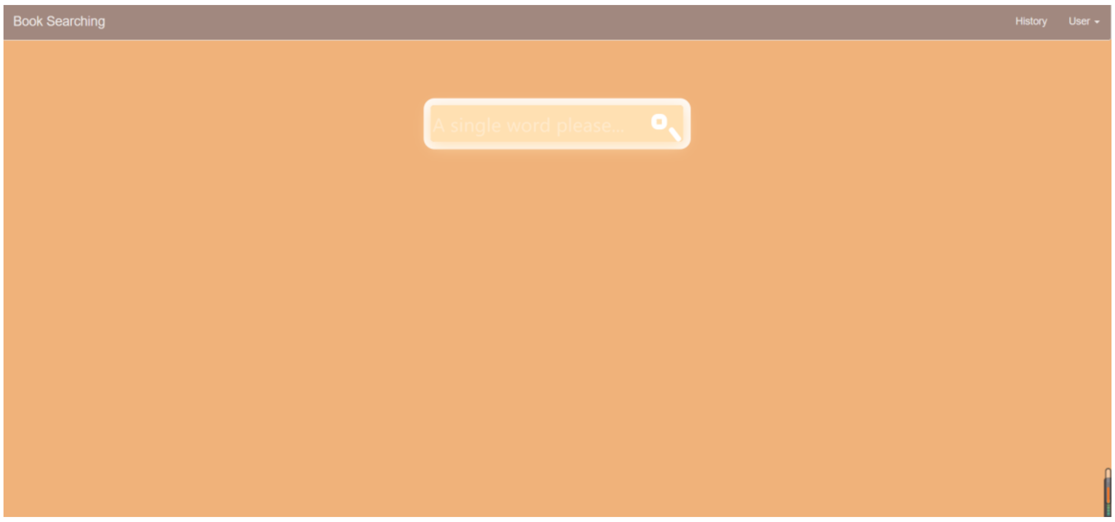
After we run the program, we get the result for ranking. The first column is the words appear in the documents. Following, it is the list of books with the word appears in it and this list is sorted in descending order.



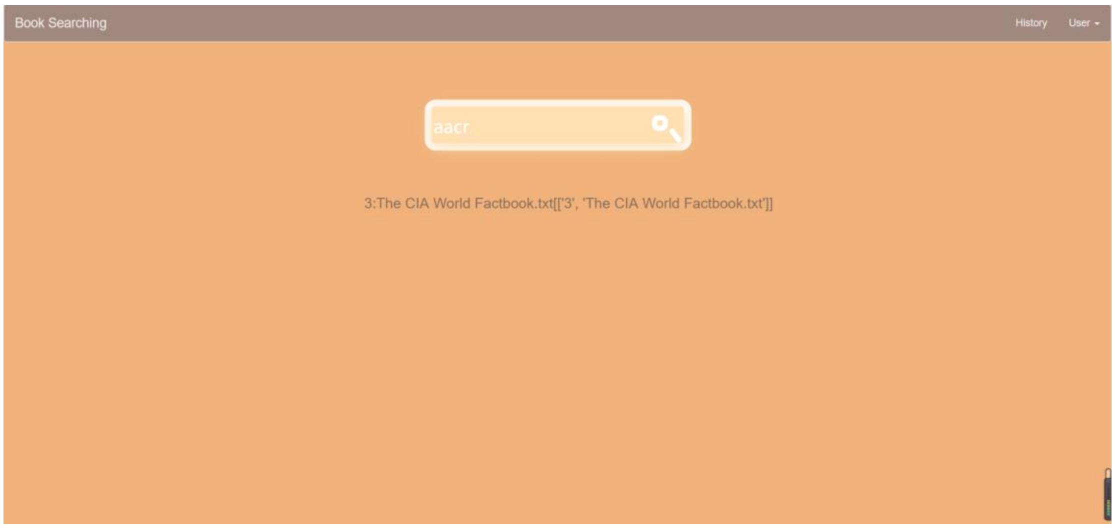
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
39	aargau	The CIA W	Great Men	The Circle of Knowledge A Classified Simplified Visualized Book of Answers.txt																		
40																						
41	aarhuus	The Circle of Knowledge A Classified Simplified Visualized Book of Answers.txt																				
42																						
43	aarnikotka	Niilo Bonpoika Sture Kuninkaankruunu Finnish.txt																				
44																						
45	aarniotaan	The fair maid of Perth Finnish Finnish.txt																				
46																						
47	aaron	The Gramr	History of '	Memoir Co	Final Repo	Great Men	The Circle (	The Bible [	The King Ja	The King Ja	The Alemb	Memoirs o	Moral Thec	Thirty Year	US Copyrig	Expositions	Biographia	The Book c	Life of Johr	History of t	The Cathol	T
48																						
49	aarones	The Grammar of English Grammars.txt																				
50																						
51	aaronical	Life of John Coleridge Patteson Missionary Bishop of the Melanesian Islands.txt																				
52																						
53	aaronites	The King Ja	The King Ja	The Bible DouayRheims Complete.txt																		
54																						
55	aarons	US Copyrig	Biographia Scoticana Scots Worthies.txt																			
56																						
57	aarran	Biographia Scoticana Scots Worthies.txt																				
58																						
59	aarre	The fair ma	Niilo Bonpoika Sture Kuninkaankruunu Finnish.txt																			
60																						
61	aarretta	The fair maid of Perth Finnish Finnish.txt																				
62																						
63	aarteen	Niilo Bonpoika Sture Kuninkaankruunu Finnish.txt																				
64																						
65	aarteeni	The fair maid of Perth Finnish Finnish.txt																				
66																						

Web Interface Design

We provide the users with a web interface to enter their queries to the system. They can type the key word about a book in the searching box:

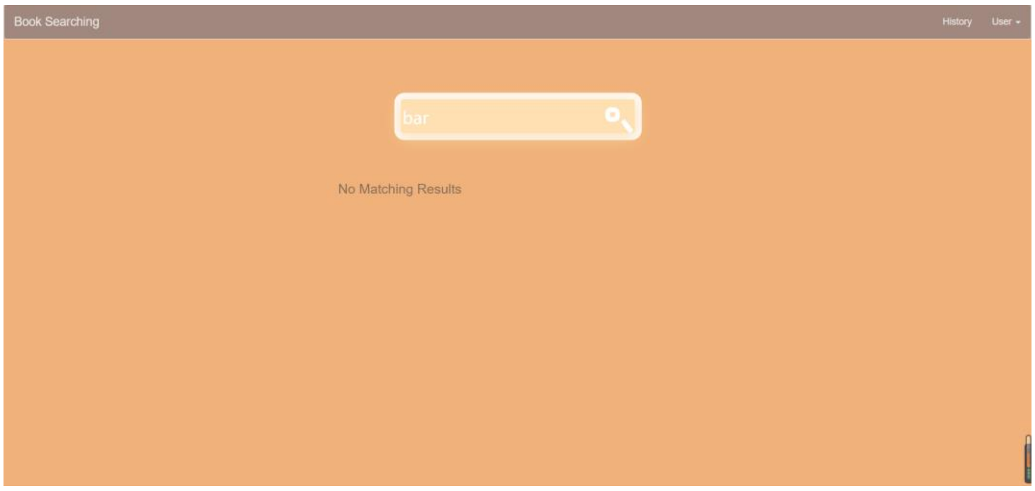


For example, if we type the word “aacr”, we will get the information as below:



3:The CIA World Factbook.txt[['3', 'The CIA World Factbook.txt']]

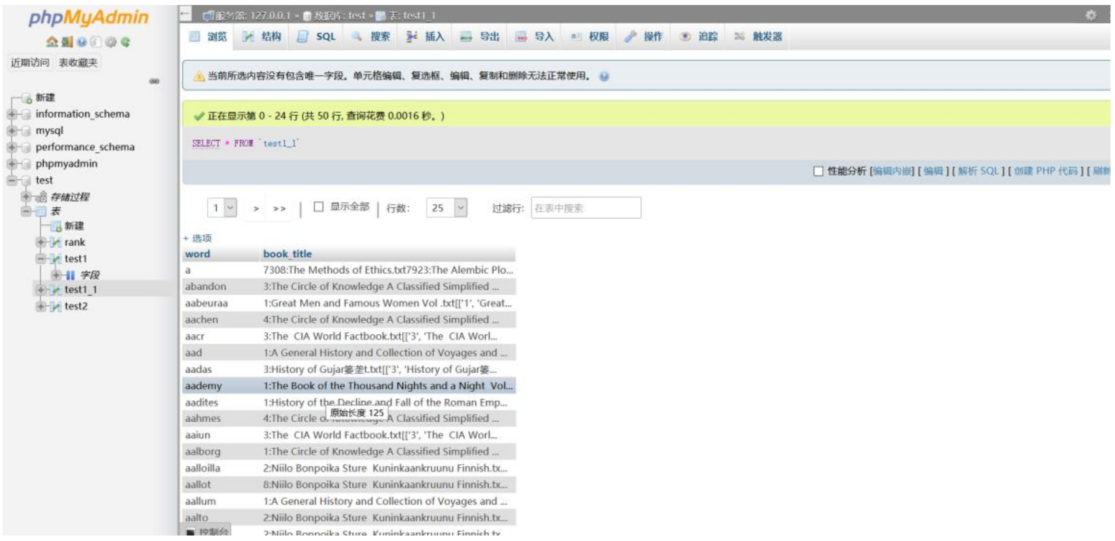
If the word that users type is not in our database, we will return the result as below:



No Matching Results

Database Design

For quicker searching, we only choose 100 words to be our test dataset of our website. We have two variables totally, which are key word and book information.
This is what our database looks like:


phpMyAdmin
近期访问 表收藏夹
新建
information_schema
mysql
performance_schema
phpmyadmin
test
存储过程
表
新建
rank
test1
字段
test1_1
test2
服务器: 127.0.0.1 » 数据库: test » 表: test1_1
浏览 结构 SQL 搜索 插入 导出 导入 权限 操作 追踪 触发器
性能分析 [编辑内嵌] [编辑] [解析 SQL] [创建 PHP 代码]
1 > >> 显示全部 行数: 25 过滤行: 在表中搜索
+ 选项
原始长度 125
控制台

当前所选内容没有包含唯一字段。单元格编辑、复选框、编辑、复制和删除无法正常使用。

正在显示第 0 - 24 行 (共 50 行, 查询花费 0.0016 秒。)

```
SELECT * FROM `test1_1`
```

word	book_title
a	7308:The Methods of Ethics.txt7923:The Alembic Plo...
abandon	3:The Circle of Knowledge A Classified Simplified ...
aabeuraa	1:Great Men and Famous Women Vol .txt[['1', 'Great...
aachen	4:The Circle of Knowledge A Classified Simplified ...
aacr	3:The CIA World Factbook.txt[['3', 'The CIA Worl...
aad	1:A General History and Collection of Voyages and ...
aadas	3:History of Gujar篓垄t.txt[['3', 'History of Gujar篓...
aademy	1:The Book of the Thousand Nights and a Night Vol...
aadites	1:History of the Decline and Fall of the Roman Emp...
aahmes	4:The Circle o[illegible] A Classified Simplified ...
aaiun	3:The CIA World Factbook.txt[['3', 'The CIA Worl...
aalborg	1:The Circle of Knowledge A Classified Simplified ...
aalloilla	2:Niilo Bonpoika Sture Kuninkaankruunu Finnish.tx...
aallot	8:Niilo Bonpoika Sture Kuninkaankruunu Finnish.tx...
aallum	1:A General History and Collection of Voyages and ...
aalto	2:Niilo Bonpoika Sture Kuninkaankruunu Finnish.tx...
	2:Niilo Bonpoika Sture Kuninkaankruunu Finnish.tx

Strength and Weakness

Strength:

1. Simple and intuitive interface;
2. Easy for users to use and get information.

Weakness:

1. Irregular arrangement of book information;
2. Users can only get the name information of books.