



SHANGHAI JIAO TONG UNIVERSITY

IEEE CLASS

INTRODUCTION TO ELECTRICAL ENGINEERING

The Report For Final Project

Author:

Liu Yuwei

Mao Weibo

Yang Nianzu

Zhao Ziheng

Student Number:

517030910295

517030910296

517030910301

517030910304

June 24, 2018

Contents

1	Abstract	2
2	Presentations of Our Work	3
3	Analyses of Codes	6
3.1	The Over-all Style of Our Websites	6
3.2	home.php	6
3.2.1	The Design of “Home” Page	6
3.2.2	How To Search For The Object I Want	6
3.3	author.php	7
3.3.1	The Integrated Layout of “author” Page	7
3.3.2	The Mentor-Student Relationships Graph	9
3.4	paper.php	14
3.4.1	Paper Recommendation	14
3.5	conference.php	15
3.5.1	Result Demonstration	15
3.5.2	Basic Information	16
3.5.3	The Chart	16
3.5.4	Paper Recommendation	18
3.6	JS/CSS/Bookstrap	21
4	Additional Functions And Data Analysis	24
4.1	Statistics of Time Consuming	24
4.2	Relationship Map for 2,000 Authors	25
4.3	Useful Websites	26
5	Summary and reflection	27
5.1	Summary	27
5.2	Reflection	27
6	Acknowledgement	28

1 Abstract

In this final lab, we are aimed to integrate all the functions that we have realized before and add some new ideas to the final work.

We first conceive the functions that we need to add for each page. We realize the selective searching function in our homepage with an integrated searching box; We create a page to demonstrate a paper's information and another page to demonstrate a conference's information; We add a dynamic effect to the demonstration of searching result for conferences; We realize the function of turning pages when demonstrating the searching results for authors or papers; We add recommendations for papers in conference page and paper page; We draw a graph showing the papers published every year in conference page; We draw a graph showing the authors' relationships in author page and another showing the mentorstudent relationships; Besides, we draw an overall graph to show all the authors' relationships.

We then build up the connections between pages. We assign hyperlinks to most of the paper's titles, author's names and conferences' names. So the visitor can switch the page from one context to another freely; We also beautify all the pages and coordinate their styles to a compact but goodlooking one.

Finally, we carry out an analysis about the time consumed to open a page and a summary on the whole lab. We also reflect on several points that still need improvement.

In this report, we skip the parts done in previous exercises. We mainly introduce and explain the new elements in our website in the final lab. Also, some result demonstrations are hard to realize in a .pdf file so we leave them in our ppt.

2 Presentations of Our Work

First of all, we will show you the framework of our websites.

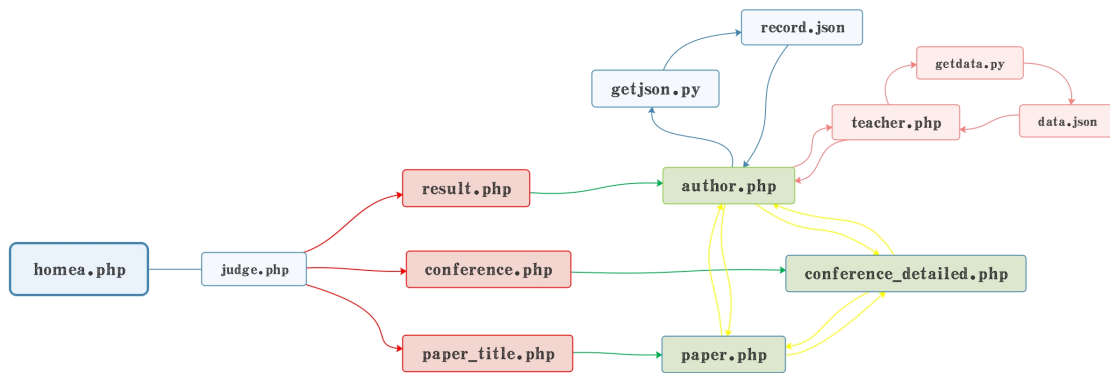


Figure 1: *Framework*

As you can see, we construct different websites and more importantly—we add complex connections to each website, which means that you can easily jump from one website to another.

This presentation will starts from the *home page*.

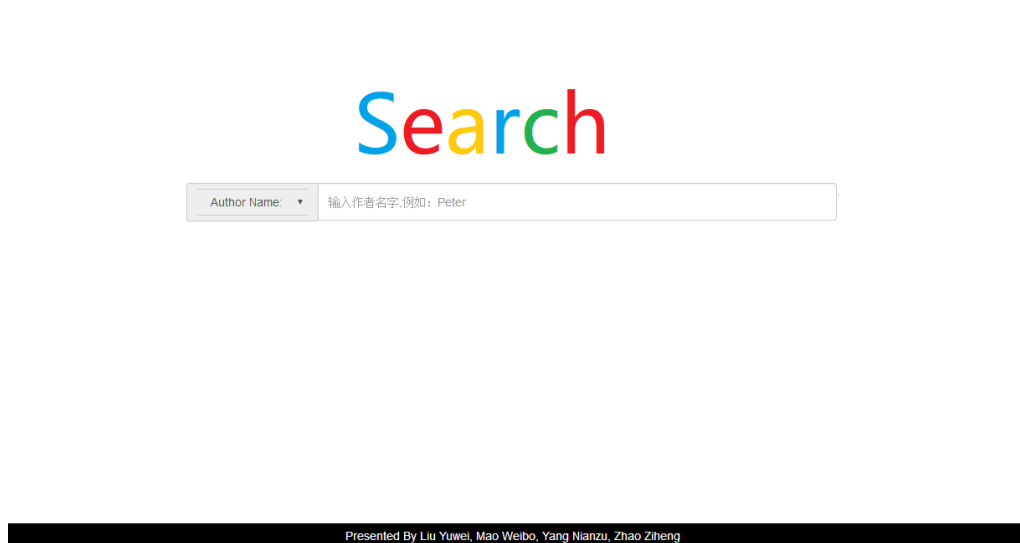


Figure 2: *Our Home Page*

If we enter ‘*joa*’, then we will get all the authors whose name contains ‘*joa*’. Also we can search for conferences or papers, we will show it later.

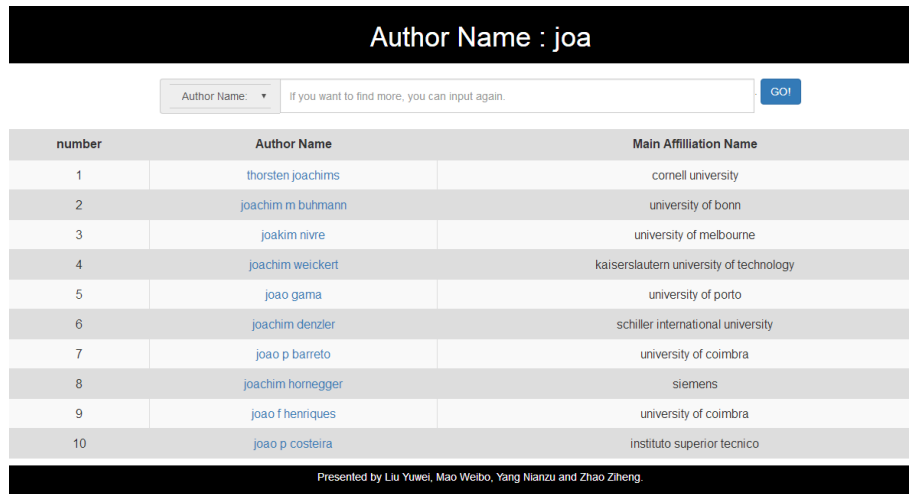


Figure 3: 'joa'

Then, if we click the specific name, we will get the detailed information of this author.



Figure 4: 'joachim m buhmann'

You can also click to see the mentor relationship map for this author.

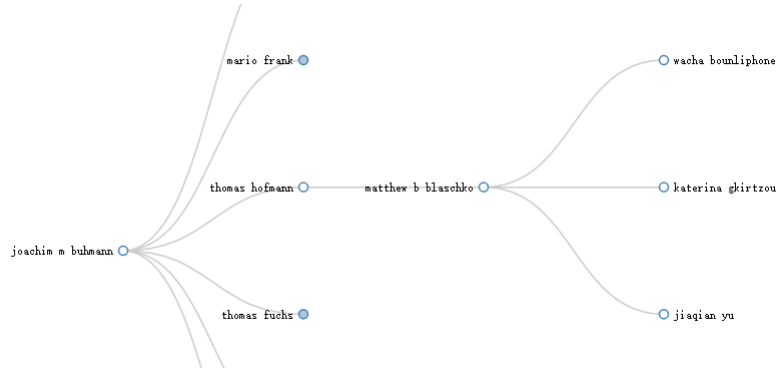


Figure 5: *mentor relationship*

Similarly, you can search for different papers, conferences or just click to goto another website.

Title : Pascal

Author Name: If you want to find more, you can input again.

number	title	Author Name	Affiliation	Publish Year
1	the 2005 pascal visual classes challenge	olivier chapelle jan radu mehmet demir	IONL	2005
2	the third pascal recognizing textual entailment challenge	ido dagan bernardo magalhães dario gonzalez	ACL	2007
3	the pascal visual object classes challenge 2006 results	chris wilfong lu wei gao mark norouzzadeh	ECCV	2006
4	reconstructing pascal voc	juliano augusto carreira roberto aguiar sara riccardi	CVPR	2014
5	the pascal challenge on grammar induction	douwe grefling hans-sabbe joost graen	NAACL	2012

Paper : toward an architecture for never ending language learning

Publish Year : 2010

Conference Name : AAAI

Authors :
1 justin betteridge
2 bryan kiesel
3 tom m mitchell
4 burr Settles
5 andrew carlson
6 astyrum r hruschka

Downloaded by Liu Yuwei, Mao Weibo, Yang Nianzu, Zhao Ziheng

Figure 6: *Search for Papers*

Conference : A

Conference : AAAI

Welcome to AAAI!

Founded in 1979, the Association for the Advancement of Artificial Intelligence (AAAI) honors the American Association for Artificial Intelligence (AIAA) scientific projects and the scientific understanding of the machine learning thought and intelligence behavior and their applications in various fields. AAAI aims to promote research in, and responsible use of, artificial intelligence. AAAI also aims to increase public understanding of artificial intelligence, improve the visibility and training of AI practitioners, and provide guidance for research planning and funding opportunities, the importance and potential of current AI developments and future directions.

5-Year AAI Index

Recommendation

Bestest Papers:

1. toward an architecture for never ending la... —Refered Times:99
2. learning distributions for information extr... —Refered Times:96
3. a new method for solving hard satisfiability... —Refered Times:88
4. statistical parsing with a context free gr... —Refered Times:82
5. pushing the envelope parsing prepositions... —Refered Times:75

Latest Papers:

1. logical foundations of assertion connecti... —Published Year:2016
2. filtering with logic program and its appl... —Published Year:2016
3. ontology mediated queries for social data... —Published Year:2016
4. discovering user attribute syntactic diff... —Published Year:2016
5. active inference and dynamic gaussian baye... —Published Year:2016

Figure 7: *Search for Conferences*

3 Analyses of Codes

3.1 The Over-all Style of Our Websites

Considering the fact that we are required to build a website which serves academically searching purpose, we chose black and white as the main colour of our websites. Therefore, our websites will look more formal and more academic. At the same time, the results you are searching for will stand out with the background of black and white. And we beautify our websites with the help of JS , CSS and Bookstrap.

3.2 home.php

3.2.1 The Design of “Home” Page

In the beginning, we tried to learn from some designs of homepage provided on the Internet. Nevertheless, these provided on the Internet are rather gaudy. They are not suitable for our academic websites. In addition, we also took it into consideration that many other students may apply these examples provided online mechanically. Therefore, we decided to design our homepage on our own in the end.

We imitated the “Google Scholar”. Its homepage is concise and clear. So we used some graphics softwares to serve our purpose. We designed a logo “Search” which is made up of several colours. At first sight, there is no difference between ours and Google’s. And below the logo is a form where you can input information for further search. What’s more, in order to make our websites close to official websites, we claim our copyrights at the bottom of the homepage.

3.2.2 How To Search For The Object I Want

We actually hide a *.php* ,whose function is leading us to the page we want. The basic idea is easy to understand—we first get two paraments from the home.php. And based on the first parament, we decide the direction while the second is the parament we need to transport.

Since we use such a direct and easy method, we can add some painted eggs here (such as searching for xinbing or poet...)

The main codes in the judge.php is as bellow.

```
1 <?php
2 $jud=$_GET[ 'aaa' ][0];
3 $name=$_GET[ 'id' ];
4 if ($name=='xinbing'){
5     header( 'Location: https://baike.baidu.com/item/%E7%8E%8B%E6%96%B0%E5
6         %85%B5/7083932?fr=aladdin' );
7 }
8 else{
9     if ($name=='poet'){
10         header( 'Location: poet.php' );
11     }
12     else{
13         if ($jud=='A'){
14             header( 'Location: result.php?key=' . $name );
15         }
16         else{
17             if ($jud=='T'){
18                 header( 'Location: paper_title.php?ppid=' . $name );
19             }
20             else{
21                 header( 'Location: conference.php?id=' . $name );
22             }
23         }
24     }
25 }
26 ?>
```

Listing 1: judge.php

As you can see above, we use a judge.php to judge which page we need to go.

3.3 author.php

3.3.1 The Integrated Layout of “author” Page

The arrangement of “author” page is much more complicated than that of “home” page. In this page, we need to divide the whole page into different parts. Each part is responsible for demonstrating different information to users.

First, we should make it clear that what functions the “author” page ought to be equipped with. This page should present the relationship graph and detailed information about the paper user searches for. Besides, it also should present a mentor-relationship graph. However, given that this graph is too large. It will affect

the overall arrangement of “author” page. Therefore, we decided to only provide a hyperlink for mentor-relationship graph in this page. User can view this graph by clicking on this hyperlink. Then it will turn to another page where the mentor-relationship graph is presented.

Then it’s time for us to consider how to cope with the remained parts. We intended to divide this page into five parts. The first part is placed at the head of the page. It will reveal the current author’s name which you are searching for. The second part is just below the first one. For the convenience of users, we also place a form in this part. So if the user want to search for other information, he or she can just realize his or her goal in this page. They don’t need to get back to the homepage. The third and the fourth parts are in the same horizontal line. That means we should learn to divide the website into two columns. The method can be learned from the Internet. And the related codes will be demonstrated as below. The left column is responsible for presenting the relationship graph and providing the hyperlink for mentor-relationship graph. And the right column will presents the searching results in a table. The last part is used to claim copyright again.

Here are related codes of division. They are demonstrated as below.

```
1 <style>
2 #header {
3     background-color : black ;
4     color : white ;
5     text-align : center ;
6     padding : 5px ;
7 }
8 #nav {
9     line-height : 30px ;
10    background-color : white ;
11    height : 400px ;
12    width : 400px ;
13    float : left ;
14    padding : 5px ;
15 }
16 #section {
17    width : 870px ;
18    float : left ;
19    padding : 10px ;
20 }
21 #footer {
22    background-color : black ;
23    color : white ;
24    clear : both ;
25    text-align : center ;
26    position : absolute ;
27    top : 650px ;
28    left : 375px ;
```

```
29 padding:5px;  
30 }  
31 </style>
```

Listing 2: Division

This method can be learn online. But the codes online can't be applied to our websites directly. We need to find suitable size for different parts. The first part is claimed as "header". And the third one is "nav". The fourth one is "section". It's the main part. The last part is claimed as "footer". What's more, we want the last part to be always fixed at the bottom of the page. Therefore, we set its position as the kind named "absolute".

The arrangement of this page is the most complicated. So we won't introduce other pages' arrangement again. We only take this one for an example. And further details of beautification will be illustrated in the JS/CSS/Bookstrap part.

3.3.2 The Mentor-Student Relationships Graph

To make the author's page looks more comfortable, we leave the mentor-student reation graph in another subpage. Here we place a link in author.php to turn to the graph.

```
1 <?php  
2 echo "<h2><a href=\"teacher.php?id=\".$name.\">>Mentor Relationship </  
3 </h2>";  
4 ?>
```

Listing 3: Add Links

Then, in teach.php, we use the js plugin and code downloaded online to help us draw a graph. Here is the website for conference:

<https://bl.ocks.org/anonymous/ab8d7f85cca6f745a107>

Basically, it realizes the function of drawing a tree graph, folding and unfolding the nodes. According to the insturction, we bulid up the tree graph and then, we have to process the data we have from database into a json form.

Here we send the author's id to a .py file and use it to help us.

```
1 #in teacher.php  
2 unset($out);
```

```
3 $c = exec("getdata.py {$name}", $out, $res);
4
5 #in getdata.py
6 id=sys.argv[1]
```

Listing 4: Call *.py* file

As we will explain later, we create a new table in our database to store every mentor-student relationship. So after we get the author's id, we search it in the table to find all its students' id.

```
1 sql = "SELECT * FROM mentors\
2     WHERE AuthorID1='%s'" % (id)
```

Listing 5: Database Codes

The form of data that we want is a overall dictionary represent the root author in this page, and it contains her or his name and a list in which every subdictionary represent one of her or his student. In this structure, the dictionary can have numerous nested layers but to make it compact, we at almost let three layers occur in the graph so there is three layer of loop to fill the overall dictionary.

For every author in a layer, we try to get its name (string) and its students (list). In case it has no student, we use a errorchecking system to directly give it an empty list. After we fill in all the author's information (dictionary) in this layer, we return it to the upper layer as a list contains all the dictionarys. And we take it as the students' information of one of the authors in upper layer. Finally in the top layer, we get all the root author's students' information in a list called final, and together with her or his name, we store them in a dictionary called finala and that is the data we need.

Note that db.rollback is to make sure we can withdraw the operation if it will do damage to the data in the database.

```
1 #layer1
2 try:
3     cursor.execute(sql)
4     results = cursor.fetchall()
5     final = []
6     for row in results:
7         tmp= row[2]
8         na=row[3]
9         name=row[1]
10        sql2 = "SELECT * FROM mentors\
11            WHERE AuthorID1='%s'" % (tmp)
12 #layer2
13     try:
14         cursor.execute(sql2)
```

```
15         results2=cursor.fetchall()
16         tmppp = []
17         for row2 in results2:
18             tmp2=row2[2]
19             na2=row2[3]
20             sql3="SELECT * FROM mentors\
21             WHERE AuthorID1='%s'" % (tmp2)
22 #layer3
23         try:
24             cursor.execute(sql3)
25             results3=cursor.fetchall()
26             tmpp=[]
27             for row3 in results3:
28                 tmp3=row3[2]
29                 na3=row3[3]
30                 tmp4={
31                     "name":na3,
32                 }
33                 tmpp.append(tmp4)
34         except:
35             db.rollback()
36             tmp5={
37                 "name":na2,
38                 "children":tmpp
39             }
40             tmppp.append(tmp5)
41     except:
42         db.rollback()
43     tmp6={
44         "name":na,
45         "children":tmppp
46     }
47     final.append(tmp6)
48 except:
49     db.rollback()
50
51 finala={
52     "name":name,
53     "children":final
54 }
```

Listing 6: Part of *.py* Codes

Finally we turn it into json data and write it in a file which can be read by teacher.php to draw the graph accordingly. Since everytime we write data in the file it will cover the previous one, they won't come into conflict with each other.

```
1 #in getdata.py
```

```
2 with open("data.json","w") as f:
3     json.dump(finala ,f ,indent=2)
4
5 #in teacher.php
6 d3.json("data.json", function(json) {
7     .....
8 });
```

Listing 7: Data Transport

Here is how the data looks like in data.json:

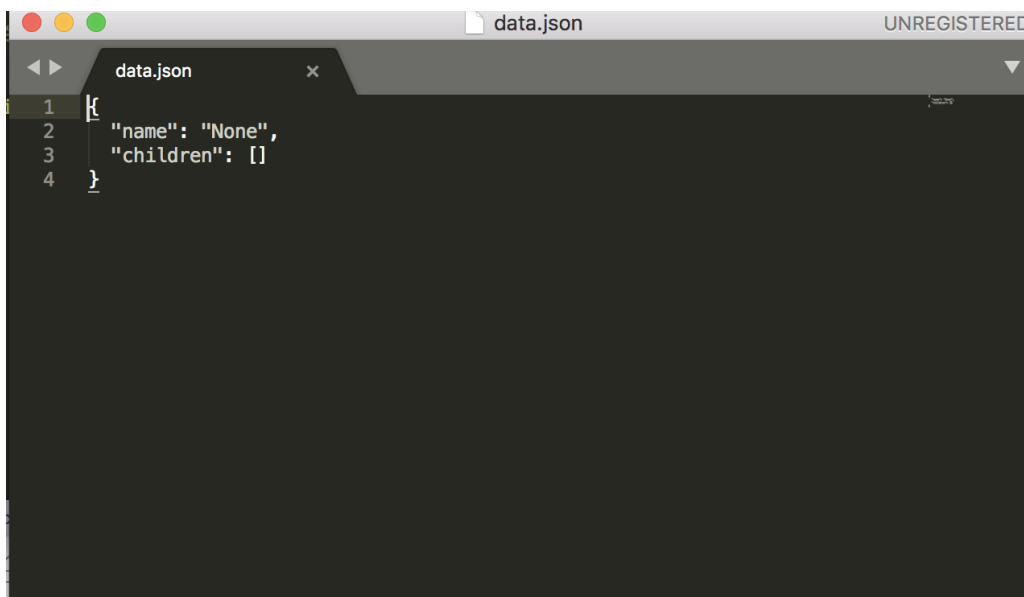


Figure 8: *Json Data*

Then we can draw out the mentorstudent graph like this:

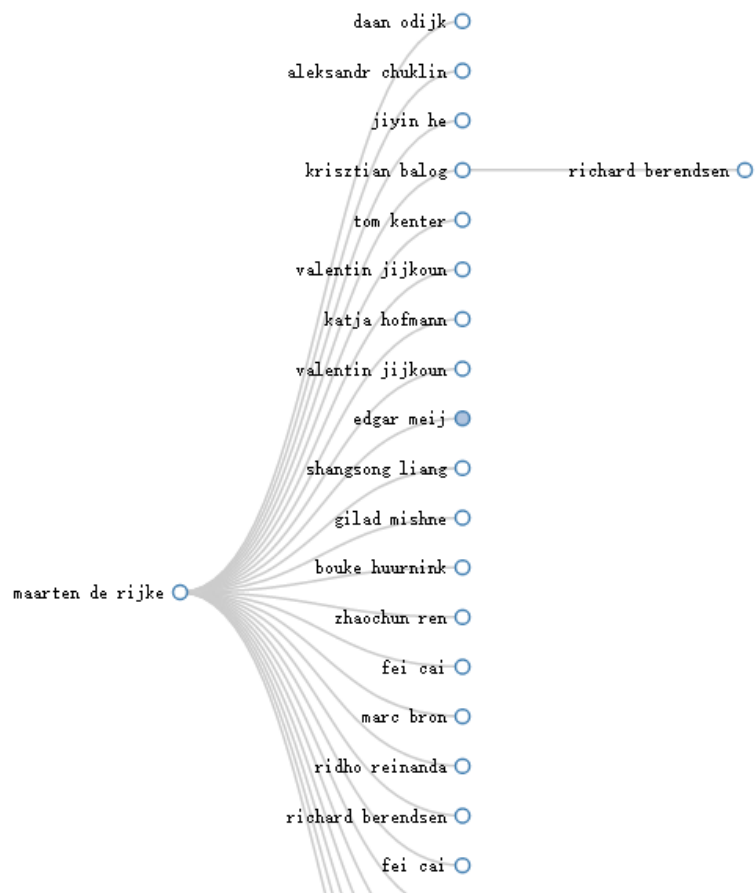


Figure 9: Mentor-Student Relationship Graph

3.4 paper.php

3.4.1 Paper Recommendation

Here, we recommend the most relative papers to you, which has the same authors or similar titles. It is not such a complex work—if you have the key word for each paper. And the following is the result.

The screenshot displays a web interface for paper search results. At the top, a black header bar contains the text "Paper : toward an architecture for never ending language learning" in white. Below this, the interface is divided into two main columns. The left column lists the search criteria: "Publish Year : 2010", "Conference Name : AAAI", and "Authors : 1.justin betteridge, 2.bryan kisiel, 3.tom m mitchell, 4.burr settles, 5.andrew carlson, 6.estevam r bruschka". The right column, titled "Recommendation", lists ten related papers with their titles and authors, such as "1.coupling semi supervised learning of categ...----Common Author: justin betteridge" and "2.javelin iii cross lingual question answeri...----Common Author: justin betteridge". At the bottom of the interface, a small footer reads "Presented By Liu Yuwei, Mao Weibo, Yang Nianzu, Zhao Ziheng".

Figure 10: The Demonstration of Paper Search Result

About the detailed solutions, we will show you in the *conference.php*, which also has the paper recommmandation part.

3.5 conference.php

3.5.1 Result Demonstration

Since the number of conferences is relatively less than authors' or papers' so it will be a dull page if we plainly show them in table. So here we add a animative effective to the tags of each candidate and make the character lager and so on to make it more beautiful.

```
1 function table($rows){
2     echo "<center>";
3     echo "<div id='section'>";
4     global $names, $sids, $time;
5         echo "<div id='navigation-block'>";
6
7         echo "<ul id='sliding-navigation'>";
8
9         for ($out=0;$out<$time;$out++){
10            echo "<li class='sliding-element'>".<font size='12px';
11            color='white'>".<a href="\conference_detailed.php?id=" . $sids[$out].
12            "\>";
13            echo $names[$out].</font>".</li>";
14        }
15    echo "</ul>".</div>";
16    echo "</center>";
17 }
```

Listing 8: Animative Effective

Here is how it looks but the animative effective can only be demonstrated in our ppt.

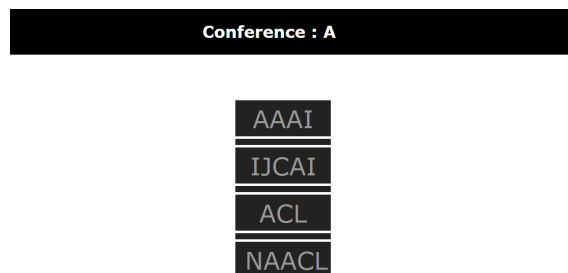


Figure 11: The Demonstration of Conferences Search Result

3.5.2 Basic Information

The basic information of a conference includes a brief introduction and its official website. The introduction is derived online and the website is found accordingly. We store them in form of string variables and echoes them according to the conference's id we receive.

```
1 if ($name=='AAAI') {
2     $introduction=$AAAI;
3     $web="https://www.aaai.org/";
4 }
5 <div id="section">
6 <?php
7 echo "<h2>Welcome to ".$name."</h2>";
8 echo "<p align='justify'>".$introduction."</p>";
9 echo "<br>";
10 echo "<a href='".$web."'>>Learn more</a>";
11 ?>
12 </div>
```

Listing 9: Concrete Codes of Using js and css

3.5.3 The Chart

To let the reader get a rough idea of how the conference develop recently, we draw a diagram to show how many papers were published each year from the year 2011 to 2016. First we get the data (or more exactly, count the papers according to the publish year) from our database. Then use implode function to link them as a string and then use echo to place them in a list and assign it to variable called data. And we also prepare the coordinates in a list.

```
1 $namea=array();
2 for($i=2011;$i<2017;$i++){
3     $sql2="SELECT PaperPublishYear FROM papers WHERE ConferenceID='Sid'
4     And PaperPublishYear='$i'";
5     $result2=mysqli_query($conn,$sql2);
6     $count=0;
7     while($row2=mysqli_fetch_array($result2)){
8         $count++;
9     }
10    array_push($namea,$count);
11 }
12 $final=implode(',',$namea);
13 echo "var data =['".$final."'];";
```

```
13 var xinforma = [ '2011', '2012', '2013', '2014', '2015', '2016' ];
```

Listing 10: Codes for Charts

Then we prepare the canvas, background, conceive the frame about the height and width and then, draw vertical and horizontal line according to make the frame visible.

```
1 #the canvas
2 var a_canvas = document.getElementById( 'a_canvas' );
3 var context = a_canvas.getContext( "2d" );
4 var gradient = context.createLinearGradient( 0,0,0,300 );
5 context.fillStyle = gradient;
6 context.fillRect( 0,0,a_canvas.width , a_canvas.height );
7 #the frame
8 var realheight = a_canvas.height - 15;
9 var realwidth = a_canvas.width - 40;
10 var grid_cols = data.length + 1;
11 var grid_rows = 4;
12 var cell_height = realheight / grid_rows;
13 var cell_width = realwidth / grid_cols;
14 context.lineWidth = 1;
15 context.strokeStyle = "#a0a0a0";
16 #draw it out
17 context.moveTo(0, realheight);
18 context.lineTo(realwidth, realheight);
19 context.moveTo(0, 20);
20 context.lineTo(0, realheight);
21 context.lineWidth = 1;
22 context.strokeStyle = "black";
23 context.stroke();
```

Listing 11: Codes to Create the Frame

Then we draw the diagrams in the graph. We input the number of paper and convert them to the y value of every diagram. Then we find its x value according to the publish year. Thus, the location and height of every diagram is fixed.

```
1 var max_v = 0;
2 for( var i = 0; i < data.length; i++){
3     if ( data[i] > max_v ) { max_v = data[i] };
4 }
5 max_vmax_v = max_v * 1.1;
6 var points = [];
7 for( var i=0; i < data.length; i++){
8     var v= data[i];
9     var px = cell_width * ( i + 1);
```

```
10 var py = realheight - realheight*(v / max_v);
11 points.push({"x":px,"y":py});
12 }
```

Listing 12: Codes to Create the Chart 2

Finally we draw the diagram out according to its x and y value and add some illustration to the graph includes its title, coordiantes name and so on.

```
1 for(var i in points){
2   var p = points[i];
3   context.beginPath();
4   context.fillStyle="green";
5   context.fillRect(p.x,p.y,15,realheight-p.y);
6   context.fill();
7 }
8 for(var i in points){
9   var p = points[i];
10  context.beginPath();
11  context.fillStyle="black";
12  context.fillText(data[i], p.x + 1, p.y - 15);
13  context.fillText(xinforma[i],p.x + 1,realheight+12);
14  context.fillText('Year',realwidth,realheight+12);
15  context.fillText('Number Of Papers',0,10);
16 }
```

Listing 13: Codes to Create the Chart 3

3.5.4 Paper Recommendation

Here, we recommend the hottest (which is quoted for the most times) and the latest 5 papers of a conference. Previously, we find out the 10 papers of each conference and store their ids in the conference table so when we need them, we can make the website run faster than deriving them when the website is loading. This part will be later explained in detail.

Then in conference_detailed.php we derive the ids and their names. And we also find the exact time that how many time it has been quoted and the publish year to demonstrated along with the title. The code almost repeats the method of deriving and processing data so we skip the explanation.

```
1 $sql1=" SELECT ConferenceName,Hot,New FROM conferences WHERE
   ConferenceID='$id' ";
2 $result1=mysqli_query( $conn, $sql1 );
3 $row1=mysqli_fetch_array( $result1);
```

```

4 $name=$row1["ConferenceName"];
5 $hot=$row1["Hot"];
6 $new=$row1["New"];
7 $hot_paperids=explode(' ', $hot);
8 $new_paperids=explode(' ', $new);
9 $hot_papernames=array();
10 $new_papernames=array();
11 $hot_referred=array();
12 $new_year=array();
13 for ($i=0; $i < 5; $i++) {
14     $paperid=$hot_paperids[$i];
15     #echo $paperid;
16     $sql2="SELECT Title , Referred FROM papers WHERE PaperID='$paperid'";
17     ;
18     $result2=mysqli_query( $conn , $sql2 );
19     $row2=mysqli_fetch_array( $result2);
20     array_push($hot_papernames, $row2["Title"]);
21     array_push($hot_referred, $row2["Referred"]);
22
23     $paperid=$new_paperids[$i];
24     #echo $paperid;
25     $sql3="SELECT Title , PaperPublishYear FROM papers WHERE PaperID='
26     $paperid'";
27     $result3=mysqli_query( $conn , $sql3 );
28     $row3=mysqli_fetch_array( $result3);
29     array_push($new_papernames, $row3["Title"]);
30     array_push($new_year, $row3["PaperPublishYear"]);
31 }

```

Listing 14: Codes For Paper Recommendation

Considering that the paper's title may be too long to demonstrate, we tailor the string and limit it to the length of 45. When it exceed the number, we cut the left part off.

```

1 for ($i=0; $i <5 ; $i++) {
2     $str=$hot_papernames[$i];
3     if (strlen($str)>45) {
4         $newstr=substr($str, 0, 42)." ...";
5     }
6     else{
7         $time=45-strlen($str);
8         $suffix=str_repeat("-", $time);
9         $newstr=$str.$suffix;
10    }
11    $hot_papernames[$i]=$newstr;
12 }
13 for ($i=0; $i <5 ; $i++) {

```

```
14 $str=$new_papernames [ $i ];
15 if ( strlen( $str ) > 45 ) {
16     $newstr=substr( $str , 0 , 42 ) . " ... " ;
17 }
18 else {
19     $time=45-strlen( $str );
20     $suffix=str_repeat( "-" , $time );
21     $newstr=$str . $suffix ;
22 }
23 $new_papernames [ $i ]=$newstr ;
24 }
```

Listing 15: Detailed Codes

Finally, we print them uniformly and add hyperlinks to their titles so the user can turn to the paper's page and check detailed information.

```
1 <div id="nav">
2 <b>Recommendation</b><br>
3 <?php
4 echo "<b>Hottest Papers:</b>" . "<br>" ;
5 for ( $i=0; $i < 5 ; $i++ ) {
6     echo "<b>" . ( $i+1 ) . " " . "</b>" . "<a href=\"paper.php?ppid=" .
7     $hot_paperids [ $i ] . "\">" . $hot_papernames [ $i ] . "</a>" . "——" . " Referred
8     Times:" . $hot_referred [ $i ] ;
9     echo "<br>" ;
10    echo "<br>" ;
11    echo "<br>" ;
12    echo "<b>Latest Papers:</b>" . "<br>" ;
13    for ( $i=0; $i < 5 ; $i++ ) {
14        echo "<b>" . ( $i+1 ) . " " . "</b>" . "<a href=" . $new_paperids [ $i ] . "\">" .
15        $new_papernames [ $i ] . "</a>" . "——" . " Published Year:" . $new_year [ $i ] ;
16        echo "<br>" ;
17    }
18 </div>
```

Listing 16: Codes to Add The Hyperlinks

The result can be seen as follow.

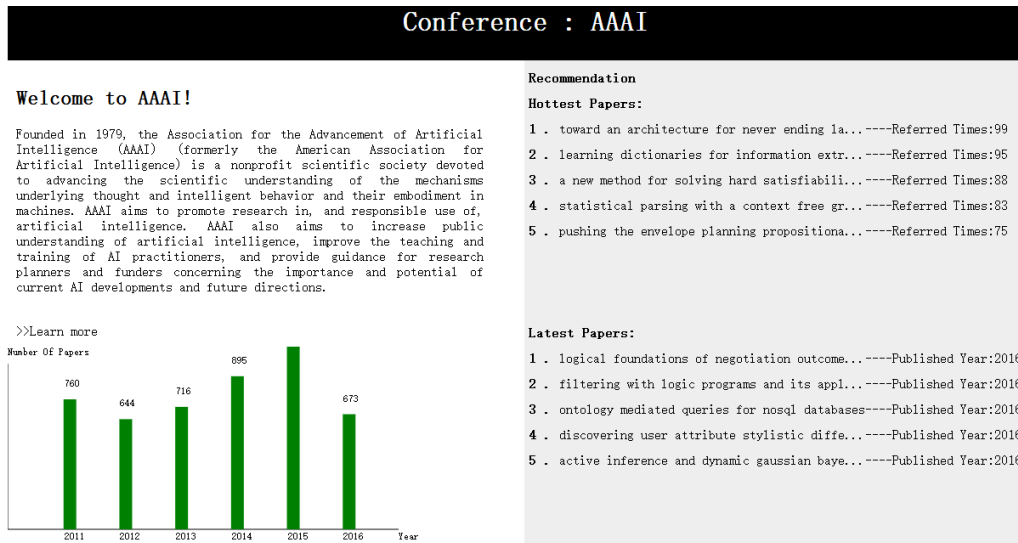


Figure 12: Conference Page

3.6 JS/CSS/Bookstrap

JS , CSS and so on are universally employed in desinging websites. They can beautify our websites. We use them many times in our pages. So it’s difficult and unnecessary for us to give introductions to all of them. Therefore, we will demonstrate some certain examples to illustrate their usages. Now, let’s take “conference” page for an example. If we input a letter “A” in the searching form in the homepage or in other pages. Then it will turn to the “conference” page. Because we use fuzzy search, then the conferences whose name contains the letter “A” will be presented. The appearance of this page is demonstrated as below.

We import a kind of JS in the head section of our “conference.php” file. It can add a sliding special effect to this page. When the cursor is moving between these conferences’ name, you can see this kind of special effect.Its script is provided as below.

```
1 <script type="text/javascript" src="sliding_effect.js"></script>
```

Listing 17: Sliding effect

Nevertheless, the special effect isn’t limited to sliding effect. When your cursor is on one conference’s name, you can find its characters will be distinguished and enlarged. CSS help us realize this goal. But it should be combined with the

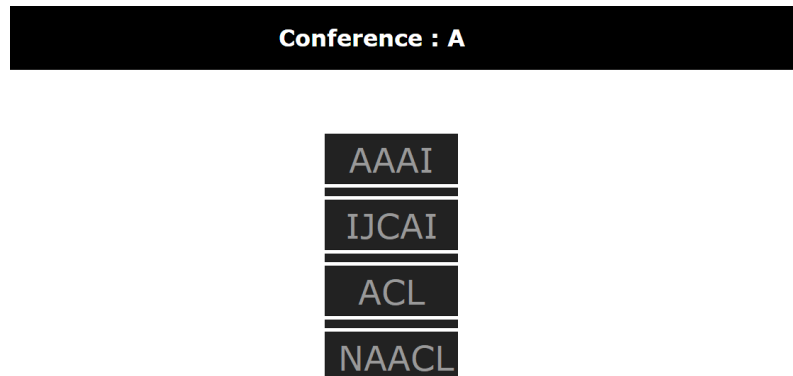


Figure 13: “conference” page

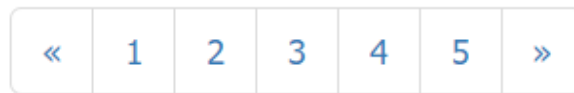
sliding_effect.js. We consult some css files online. But this part is written by ourselves completely. We created a css file named “styles.css”. And we also import it in the head section of our “conference.php” file. Its comprehensive contents will be presented in the package of all files later. In this file, we defined many types of characters. The type of the characters will be changed in response to the cursor’s location. The concrete codes of realizing this are demonstrated as below.

```
1 function table($rows){
2     echo "<center>";
3     echo "<div id='section'>";
4     global $names, $ids, $time;
5         echo "<div id='navigation-block'>";
6
7         echo "<ul id='sliding-navigation'>";
8
9         for ($out=0;$out<$time;$out++){
10             echo "<li class='sliding-element'>". "<font size='12px' ;
11             color='white'>". "<a href=\"conference_detailed.php?id=\" . $ids[$out] .
12             \"\>";
13             echo $names[$out]. "</font>". "</li>";
14         }
15     echo "</ul>". "</div>";
16     echo "</center>";
17 }
```

Listing 18: concrete codes of using js and css

In addition, we also use Bootstrap to beautify our websites. Bootstrap is based on HTML, CSS and Javascript. In a sense, it’s a combination of CSS and JS.

It's very easy to use. It makes CSS dynamic. We use it to beautify our result tables , searching forms and so on. For example, in before exercises, we are required to add two buttons named "Previous" and "Next". But these two button are not pleasing to the eye. So we ask help for Bookstrap. It contains module named "pagination-pagination-lg". You can directly use it after you import Bookstrap. Then the turning page area become more beautiful than before. The new appearance of this area is demonstrated as below. There exist many other position where we use JS ,



Presented by Liu Yuwei, Mao Weibo, Yang Nianzu and Zhao Ziheng.

Figure 14: New turning page area

CSS and Bookstrap. You can find more in the whole codes provided in addition.

4 Additional Functions And Data Analysis

4.1 Statistics of Time Consuming

We first compare the SQL speed between indexed database and none-indexed database. Actually, one of our members has done that before.

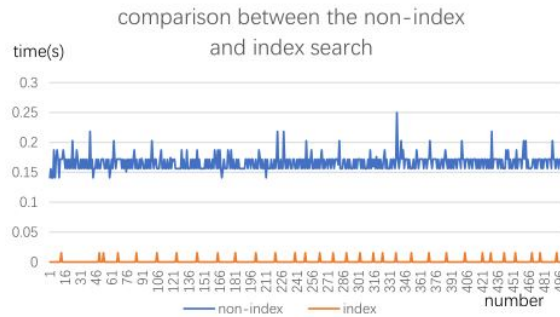


Figure 15: *Search Speed Comparison*

Inspired by the figure above, we add indexes to database when it is searched more than once. And that action does improve our speed of the website. We also collect different datas to show the response speed.

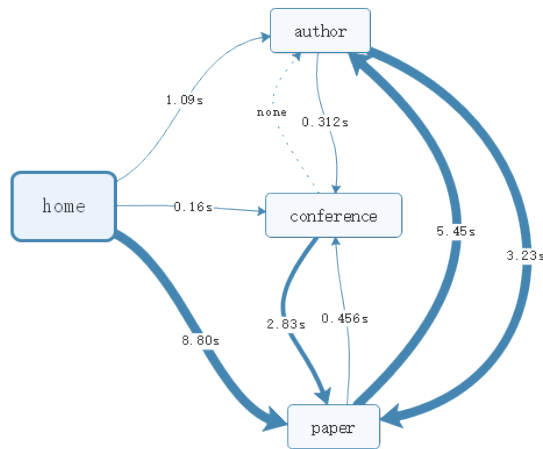


Figure 16: *Response Speed*

4.2 Relationship Map for 2,000 Authors

After finishing the main part of our project, we wonder how much authors can be derived from just one author. And we put that vagary into actions. Partly because of the complex of the code, we find that it's too time-consuming to get all the authors. For that problem, we figure out two ways to solute it. One is that we find all the relationships between the 110,000 authors, another is limit the number to 2,000. And actually, we try both. *Record-all.json* records the data we get from one author and expand him into 2,000 authors. *Record-all1.json* records all the datas.

The following records how it expand.

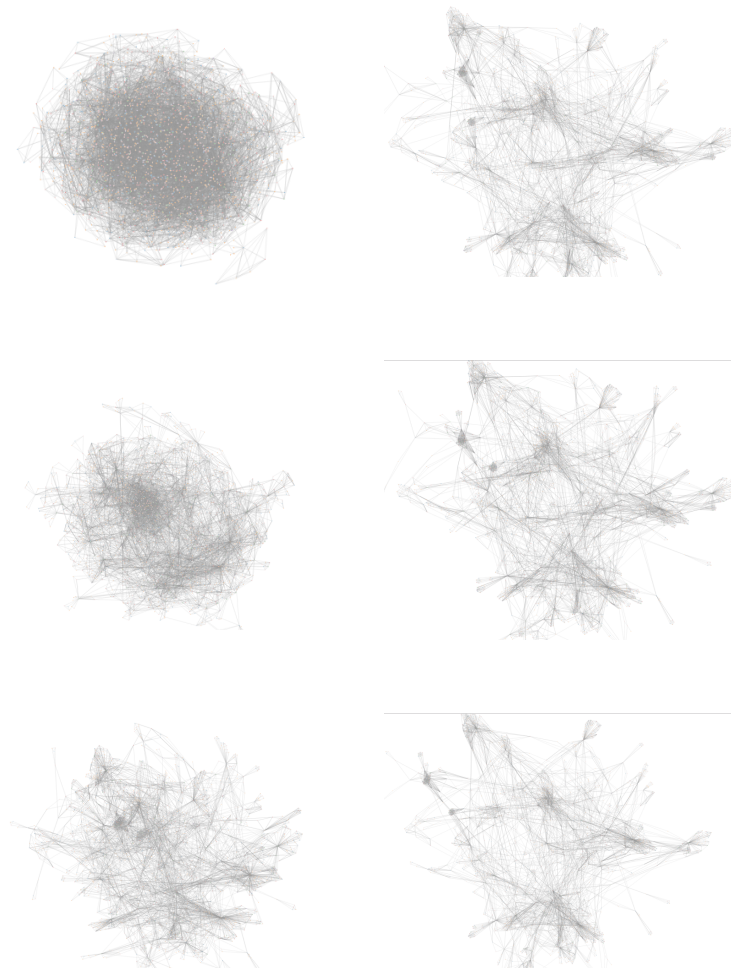


Figure 17: *Relationship Map-2,000*

For all the relationships, with the computer limited, we can not run it and get the results.....

4.3 Useful Websites

For Mentor-Relationship Map:

- <https://bl.ocks.org/anonymous/ab8d7f85cca6f745a107>
- <https://blog.csdn.net/faith306/article/details/20587365>

For JS/CSS:

- js.mobanwang.com/201107/10560.html
- js.mobanwang.com/201104/9628.html
- <http://js.mobanwang.com/201106/10407.html>

For The Different Figures:

- <https://www.jb51.net/article/113751.htm>
- <https://www.cnblogs.com/yuanke/p/4898756.html>

For Machine Learning:

- https://blog.csdn.net/qq_35576544/article/details/79547918
- <http://python.jobbole.com/81019/>

5 Summary and reflection

5.1 Summary

This project is a combination of the exercises we have finished before. Through these exercises, our websites have been equipped with basic functions. If we are arranged to finish this final project directly without the foundation of the exercises before, it's no exaggeration to say that it will rather difficult to finish it. So even though an assignment is so complicated that it seems to be insoluable, it can be divided into smaller parts. We can solve these parts one by one. Then such a troublesome problem can be solved. And through this project, we realized that teamwork is indispensable when facing such assignment. If one of us had chosen to complete this project alone, it would have cost him too much time and affect the preparations for all kinds of final examinations.

This project not only requires us to combine the achievements we attained in the exercises before, but also requires us to add more functions to our websites. Some of these functions are expansion of the exercises before, but they are more complex. Fortunately, it's not so difficult for us to realize them on the foundation for former exercises. Nevertheless, some functions are brand-new. They cost us a great deal of time to cope with them. In the end, with cooperation, we almost realized all functions required.

Futhermore, after realizing basic functions, we wanted to make our websites more user-friendly. Therefore, we polished our websites in many aspects. For example, we add a function which can recommend related articles to user when user is looking through an article. What's more, we made new attempts. The relationship graph can only presents a few authors. Nevertheless, we tried to make it can reveal authors as many as possible and we succeeded. It has been shown in former section.

5.2 Reflection

Though we has finished basic tasks, there are some aspects of our websites remaining to be developed. For instance, the searching speed is not so ideal. This problem is particularly serious in "author" page. This page need more time to load than other pages. Because this page should search considerable information in database. Even though we had added some indexes in database, this problem still

exists. Maybe we can decrease the times that the website visits database. We will try to improve it later.

Moreover, we knew many groups used elasticsearch through their presentations. However, we didn't insert it into our websites. This method has many advantages. And it can accelerate the searching speed. So we will attempt to realizing the searching function based on elasticsearch.

In addition, there still exists a gap between the appearance of our websites and that of Acemap. We need to optimize the overall arrangement of our websites later. We may redesign the overall layout and will make efforts to make all parts look more harmonious.

There are some other aspects needing to be improve in store for us. So there still a long way for us to go. It's not so easy to make a excellent website. But we will improve our websites continuously later.

6 Acknowledgement

In the end, we want to show our gratitude to Mr.Wang and tutor Jia Yuting. After this semester, we have learned rather considerable knowledge and our programming ability has been enhanced. We believe that these new knowledge will benefit us a lot later.

Thanks for your efforts!