

Final Project Report

Wang yuxin, Yu cong, Zhou qinye, Zuo yaxi

Group 9

We made a website mostly on what we learned this semester. The website includes a homepage, an author-searching page, a conference-searching page, the authors' personal pages and the conferences' pages with basic functions. The pages are linked to a database when called. Auto-complete and relationship graphs are added as advanced functions.

Contents

1	Problem Describing	2
1.1	basic part	2
1.2	Improve part(Direction for reference)	2
2	Problem Analysis and Solution	2
2.1	basic part	2
2.1.1	part 1—search for paper title and conference	2
2.1.2	part 2—add paper and conference pages in the website	5
2.1.3	part 3—paging funcion	6
2.2	improving part	10
2.2.1	part 1—recommend paper	10
2.2.2	part 2—beautify the web page	13
2.2.3	part 3—the graph of the author relationship	17
3	Results Display	19

1 **1. Problem Describing**

2 *1.1. basic part*

3 Integrate the website system completed before, improve the function and
4 optimize the interface.

- 5 • Search results page increases search for paper title and conference.
- 6 • Add paper and conference pages on the website
- 7 • When displaying information, each page displays more information
8 items (such as search results and papers).List, etc., showing 10 pages
9 per page and adding the function of page break.

10 *1.2. Improve part(Direction for reference)*

- 11 • Papers recommended (such as recommendation by co authors and com-
12 mon references) are available in papers. The pages can be recommended
13 by relevant papers, and can also be read on the scholar page through
14 this scholar's thesis.The recommendation of closing the paper and so
15 on.
- 16 • Search results are no longer based on simple MySQL queries, and can
17 be checked with Solr or elasticsearch to realize the cable function.
- 18 • The tree structure of teacher relationship shows that we can expand
19 the teacher relationship tree by layer. In addition to the examination.
20 Worry about the possibility that a scholar may have multiple tutors.
- 21 • Website performance acceleration, database optimization, SQL state-
22 ment optimization, etc.

23 **2. Problem Analysis and Solution**

24 *2.1. basic part*

25 *2.1.1. part 1—search for paper title and conference*

26 Combined with the previous situation, there are totally three informa-
27 tion(author name, paper title and conference) can be searched in the home
28 page. In order to realize the goal, we add two search boxes to search the
29 paper title and conference just like what we do in the lab2.The following is
30 the realization of adding the paper title search box.

```

311 <form action="result_paper1.php" method="get">
322 <center>PaperTitle<input type="text" id="papertitle" name="
33     papertitle">
343 <input type="submit" value="Search"><br></center>
354 </form>

```

36 Then, we also need to realize the automatic complementing function. Firstly,
37 we write the javascript code in the home page like this.

```

381 <script>
392 $(document).ready(function(){
403 $('#papertitle').autocomplete({
414     source: "hint_3.php",
425     minLength: 1
436 });
447 });
458 </script>

```

46 Secondly, we should finish the code in hint3.php just like this.

```

471 <?php
482 include_once("connect.php"); //connect with the database
493 //get the information of people entering
504 $p="\ ". "%". $_GET['term']. "%\ ";
515 $con = mysql_connect("localhost", "root", "");
526 mysql_select_db("text_1", $con);
537 $it=sprintf('select title
548     from papers
559     where title like %s
560     limit 0,10;', $p);
571 $result1 = mysql_query($it, $con);
582 $data=array();
593 while($row = mysql_fetch_array($result1))
604 {
615     array_push($data, array("label"=>trim($row['title'])));
626 }
637 echo json_encode($data); //output the data in the form of json
648 mysql_close($con);
659 ?>

```

66 In addition, in order to reduce the use of the code, we write a file named
67 connect.php which is used to connect with the database. Therefore, when we
68 need to connect with the database, we just need to include the file and select
69 the database. Here is the connect.php.

```

701 <?php
712 $host="localhost";

```

```

723 $db_user="root";
734 $db_pass="";
745 $db_name="text_1";
756 $timezone="Asia/Shanghai";
767
778 $link=mysql_connect($host,$db_user,$db_pass);
789 mysql_select_db($db_name,$link);
790 mysql_query("SET names UTF8");
801
812 header("Content-Type: text/html; charset=utf-8");
823 ?>

```

83 Up to now, we finish adding the search of paper title. Similarly, we can add
84 the search of conference. Here are the code about the search of conference
85 in the home page.

```

861 <form action="result_conference2.php" method="get">
872 <center>conference<input type="text" id="conference" name="
88     conference">
893 <input type="submit" value="Search"><br></center>
904 </form>
915 <script>
926     $(document).ready(function(){
937     $('#conference').autocomplete({
948         source: "hint_2.php",
959         minLength: 1
960     });
971 });
982 </script>

```

99 Here is the code in hint2.php. It is mainly different from the hint3.php in
100 the aspect of sql statement.

```

1011 <?php
1022 include_once("connect.php"); //connect with the database
1033 //get the information of people entering
1044 $p="\ ". "%". $_GET['term']. "%\ ";
1055
1066 $con = mysql_connect("localhost","root","");
1077 mysql_select_db("text_1", $con);
1088 $it=sprintf('select conferencename
1099     from conferences
1100     where conferencename like %s;', $p);
1111 $result1 = mysql_query($it, $con);
1122 $data=array();
1133 while($row = mysql_fetch_array($result1))
1144 {

```

```

1155     array_push($data , array("label"=>trim($row['conferencename'])))
116         ;
1176 }
1187 echo json_encode($data); //output the data in the form of json.
1198 mysql_close($con);
1209 ?>

```

121 2.1.2. part 2—add paper and conference pages in the website

122 In this part,we use sql statement to find the information we need.First,
123 we get the information from home by the method of get.And search the
124 title,paperpublishyear and paperid from papers.

```

1251 $s=$_GET["papertitle"];
1262 $result = mysqli_query($con,"SELECT title ,paperpublishyear ,
127     paperid from papers
1283     WHERE title LIKE '%{$s}%' ");

```

129 Second,use the paperid we just get to search the another information and
130 output them.

```

1311 while ($row = mysqli_fetch_array($result ,MYSQLI_ASSOC) and $n<$i
132     +10)
1332 {
1343     $paper=$row['paperid'];
1354     $result1=sprintf("SELECT authorname ,paper_author_affiliation .
136         authorid from paper_author_affiliation
1375         join authors on authors.authorid=
138         paper_author_affiliation .authorid
1396         WHERE paper_author_affiliation .paperid ='%s'", $paper);
1407     $query2=mysqli_query($con , $result1);
1418     $row2=mysqli_fetch_array($query2 ,MYSQLI_ASSOC);
1429     $tmp=mysqli_query($con,"select conpaperid from conection
1430 where paperid='{ $paper}' limit 0,3");
1441     echo "<tbody><tr><td>{$row['title']}</td> ";
1484     echo "</tr></tbody>";
1495     mysqli_free_result($query2);}

```

150 Besides,the code about realization of the paging function and recommend
151 paper in the paper.php will be showed in the part 3 of basic part and the
152 improving part.

153 Similarly,we can finish the conference.php.The following is the code.

```

1541 $s = $_GET["conference"];
1552 $result = mysqli_query($con,"SELECT * FROM conferences
1563 where ConferenceName LIKE '%{$s}%' ");
1574 $row = mysqli_fetch_array($result ,MYSQL_ASSOC);
1585 $conname=$row['ConferenceName'];
1596 $conid=$row['ConferenceID'];
1607
1618 echo '<table class="table table-striped table-bordered"><tbody>
162     <tr><td>paper title</td><td>Recommended paper</td><td>
163     paperpublish year</td><td>authorname</td></tr></tbody>';
1649 $result2=mysqli_query($con,"SELECT * FROM papers
1650 where ConferenceID='{$conid}'");
1661 while($row2 = mysqli_fetch_array($result2 ,MYSQL_ASSOC))
1672 {
1683     {
1694     $paper=$row2['PaperID'];
1705
1716     $result1=sprintf("SELECT authorname,paper_author_affiliation.
172     authorid from paper_author_affiliation
1737     join authors on authors.authorid=
174     paper_author_affiliation.authorid
1758     WHERE paper_author_affiliation.paperid ='%s'", $paper);
1769     $query2=mysqli_query($con,$result1);
1770     $row3=mysqli_fetch_array($query2 ,MYSQL_ASSOC);
1781
1792     $papertitle=$row2['Title'];
1803     $ppy=$row2['PaperPublishYear'];
1814     $tmp=mysqli_query($con,"select conpaperid from conection
1825 where paperid='{$paper}' limit 0,3");
1836     echo "<tbody><tr><td> {$papertitle}</td> ".
1847         "<td>{$ppy}</td>".
1858         "<td><a href=/author11.php?id=$row3[authorid]>{$row3['
186     authorname']}</a></td>".
1879         "</tr></tbody>";

```

188 2.1.3. part 3—paging funcion

189 In this part, we will show how to realize the paging function in the re-
190 sult.php. As for the realization of paging function in other page, because the
191 implementation methods are all the same, we will not say much about it.
192 First, I create a new file named result11.php. Then, in the file, I first include
193 the jquery.

```

1941 <script src="http://cdn.static.runoob.com/libs/jquery/1.10.2/
195     jquery.min.js">

```

```
196 2 </script>
```

197 Then I define two function (previous and next) to execute the page func-
198 tion.And in this two functions,we first need to use the way of get to get the
199 data from result10.php.Second use the document.getElementById("da").innerHTML=data
200 of HTML DOM to replace the content which id="da" by data;Here is the
201 function previous:

```
202 1 function previous() {  
203 2     $(document).ready(function() {  
204 3         $.get("result10.php",function(data,status){  
205 4             document.getElementById("da").innerHTML=data;  
206 5         });  
207 6     });  
208 7 };
```

209 Besides,since there is a browser compatibility problem in the code above,so
210 I use the following code to replace the "document.getElementById("da").innerHTML=data;"
211 1 \$("#da").html(data);

212 In addition, we should sent a data(i) to the result10.php, the function of
213 i is to limit the number of query result in one page.I let my website has 10
214 record each page. so when click on the Previous button, i should decrease
215 10. And when we click the button Next, i should increase 10.And I use
216 "window.location.search" to send the i to result10.php.Here is the function
217 next():

```
218 1 var i=-9;  
219 2 function next() {  
220 3     i=i+10;  
221 4     $(document).ready(function() {  
222 5         $.get("result10.php"+window.location.search+"&i="+String(i),  
223             function(data,status){  
224 6             $("#da").html(data);  
225 7         });  
226 8     });  
227 9 };
```

228 And we should use " i " in the result10.php which is for query informa-
229 tion. It is same as the result.php in lab2 except using "i". I just get the
230 data "i" and add a conditional statement in the origon loop statement.The
231 following is the difference:

```
232 1 $i=$_GET["i"];  
233 2 $n=0;
```

```

234.3 while($row = mysqli_fetch_array($retval, MYSQL_ASSOC) and $n<$i
235     +10)
236.4 {
237.5     $n=$n+1;
238.6     if ($n>=$i)
239.7     {
240.8     $sql=sprintf('select affiliations.affiliationname ,count(*)
241         from affiliations join paper_author_affiliation on
242.9     affiliations.affiliationid=paper_author_affiliation.
243         affiliationid
244.0     where authorid="%s"
245.1     group by paper_author_affiliation.affiliationid order by count
246         (*) desc limit 1',$row['authorid']);
247.2     $retvall = mysqli_query($conn, $sql);
248.3     if(!$retvall)
249.4     {
250.5         die('can not read data: ' . mysqli_error($conn));
251.6     }
252.7     $rowl=mysqli_fetch_assoc($retvall);
253.8     echo "<tr><td> {$row['authorid']}</td> ".
254.9         "<td><a href=/author11.php?id=$row[authorid]>{$row['
255         authorname ']} </a></td> ".
256.0         "<td>{$row['count(*) ']} </td> ".
257.1         "<td>{$rowl['affiliationname ']}</td>".
258.2         "</tr>";
259.3     mysqli_free_result($retvall);
260.4     $sql='';}
261.5 }

```

Listing 1: result10.php

262 The following is the origin result.php:

```

263.1 $n=1;
264.2 while($row = mysqli_fetch_array($retval, MYSQL_ASSOC))
265.3 {
266.4     $sql=sprintf('select affiliations.affiliationname ,count(*)
267         from affiliations join paper_author_affiliation on
268.5     affiliations.affiliationid=paper_author_affiliation.
269         affiliationid
270.6     where authorid="%s"
271.7     group by paper_author_affiliation.affiliationid order by count
272         (*) desc limit 1',$row['authorid']);
273.8     $retvall = mysqli_query($conn, $sql);
274.9     if(!$retvall)
275.0     {
276.1         die('can not read data: ' . mysqli_error($conn));

```



```

277.2 }
278.3 $row=mysqli_fetch_assoc($retvall);
279.4 echo "<tr><td> {$row['authorid']}</td> ".
280.5     "<td><a href=/author.php?id={$row[authorid]}>{$row[
281     'authorname']} </a></td> ".
282.6     "<td>{$row['count(*)']} </td> ".
283.7     "<td>{$row['affiliationname']}</td>".
284.8     "</tr>";
285.9 mysqli_free_result($retvall);
286.0 $n=$n+1;
287.1 if ($n>10) break;
288.2 $sql='';
289.3 }

```

Listing 2: result.php

290 Finally, we just need to use "onclick" to connect the button with the
291 function and write a `id` which id is `da` to be the replaced object.

```

292.1 <p id="da">
293.2 </p>
294.3 <button id="button1" onclick="previous()" >Previous</button>
295.4 <button id="button2" onclick="next()">Next</button>

```

296 The following is the complete code of result11.php:

```

297.1 <!DOCTYPE html>
298.2 <html>
299.3 <head>
300.4 <meta charset="utf-8">
301.5 <script src="http://cdn.static.runoob.com/libs/jquery/1.10.2/
302     jquery.min.js">
303.6 </script>
304.7 <script>
305.8 var i=-9;
306.9 function next(){
307.0     i=i+10;
308.1     $(document).ready(function(){
309.2         $.get("result10.php"+window.location.search+"&i="+String(i),
310         function(data,status){
311.3             //document.getElementById("da").innerHTML=data;
312.4             $("#da").html(data);
313.5
314.6         });
315.7     });
316.8 };
317.9 function previous(){

```

```

3180 i=i-10;
3191 $(document).ready(function(){
3202     $.get("result10.php"+window.location.search+"&i="+String(i),
321     function(data,status){
3223         //document.getElementById("da").innerHTML=data;
3234         $("#da").html(data);
3245     });
3256 });
3267 };
3278 next();
3289 </script>
3290 </head>
3301 <body>
3312 <center>Author name: <?php echo $_GET["name"]; ?><br></center>
3323 <center>The results are as follows:<br>
3334 <p id="da">
3345 </p> </center>
3356 <center>
3367 <button id="button1" onclick="previous()" >Previous</button>
3378 <button id="button2" onclick="next()">Next</button></center>
3389 </body>
3390 </html>

```

340 Similarly, we can execute page function in other pages just doing the same
341 things in the above. So I don't talk something more.

342 2.2. improving part

343 2.2.1. part 1—recommend paper

344 At first, we recommended the other articles of the first author of the target
345 article. But the relationship between the two articles is too weak. Under the
346 advice of other students, we first find all the articles of the author of the
347 article, in which we find articles that have a reference to the target article,
348 as a recommendation article.

349 First, we set up an associated form (connection) through the python statement.
350 Define a function to find all the articles written by the author of the target
351 paper.

```

3521 def allpapers(paperid):
3532     l=[]
3543     sql1="""select authorID from paper_author_affiliation where
355     paperID="{0}" """.format(paperid)
3564
3575
3586

```

```

359 7     cursor.execute(sql1)
360 8
361 9     results = cursor.fetchall()
362 0     for row in results:
363 1         authorid=row[0]
364 2         sql2 = """select paperiD from paper_author_affiliation
365         where authoriD="{0}" and paperiD!="{0}" """.format(authorid ,
366         paperid)
367 3
368 4         cursor.execute(sql2)
369 5         results2 = cursor.fetchall()
370 6         for row2 in results2:
371 7             if row2[0] not in l:
372 8                 l.append(row2[0])
373 9
374 0
375 1     return l

```

376 Define a function to find all the quoted articles in a given article.

```

377 1 def reference(id):
378 2     l=[]
379 3     sql3="""select referenceiD from paper_reference where
380     paperiD ="{0}" """.format(id)
381 4     cursor.execute(sql3)
382 5     results3 = cursor.fetchall()
383 6     for row in results3 :
384 7         l.append(row[0])
385 8     return l

```

386 If the article A B has the same author,and A B has a reference relation-
387 ship.The definition of the two articles is related.

```

388 1 def conn(idA ,idB):
389 2     lb=reference(idB)
390 3     if idA in idB:
391 4         return True
392 5     la=reference(idA)
393 6     for i in lb:
394 7         if i in la:
395 8             return True
396 9     return False

```

397 Set up a table "conecion"

```

398 1 sql1=""" select paperiD from papers """
399 2 cursor.execute(sql1)
400 3 results=cursor.fetchall()

```

```

4014 for row in results:
4025     paperid=row[0]
4036     l=allpapers(paperid)
4047     for i in l:
4058         if conn(paperid,i):
4069             sql2 = """insert into conection(paperID,conpaperid)
4070                 values("{}","{}")""".format(paperid,i)
4081             cursor.execute(sql2)
4092             connection.commit()
4103
4114
4125 connection.close()

```

413 Query through the PHP statement directly to the table to find the recommended articles.

```

4151 $tmp=mysqli_query($con,"select conpaperid from conection
4162     where paperid='{ $paper}' limit 0,3");

```

417 Add hyperlinks to jump to the display page.

```

4181 $name_tmp=$name_tmp."<a href=/result_tmp_paper2.php?tmp_id=" .
419     $tmp2_row['paperid'].">" . $tmp2_row['title']."</a>" . "<br>";}

```

420 The display page displays the article's title,authurname and publishyear.

```

4211 while ($row = mysqli_fetch_array($result,MYSQLI_ASSOC))
4222 {
4233
4244     $paper=$row['paperid'];
4255     $result1=sprintf("SELECT authurname,paper_author_affiliation .
426     authorid from paper_author_affiliation
4276     join authors on authors.authorid=
428     paper_author_affiliation.authorid
4297     WHERE paper_author_affiliation.paperid ='%s'", $paper);
4308     $query2=mysqli_query($con,$result1);
4319     $row2=mysqli_fetch_array($query2,MYSQLI_ASSOC);
4320     $tmp=mysqli_query($con,"SELECT paper_author_affiliation .
433     authorid
4341 from paper_author_affiliation join papers on
435     paper_author_affiliation.paperid= papers.paperid
4362     where papers.paperid like '{ $paper}'");
4373     $tmp_row=mysqli_fetch_array($tmp,MYSQLI_ASSOC);
4384     $tmp2=mysqli_query($con,"SELECT papers.title ,papers.paperid
4395 from paper_author_affiliation join papers on
440     paper_author_affiliation.paperid= papers.paperid
4416     where paper_author_affiliation.authorid='{ $tmp_row['authorid
442     ]}' and papers.paperid!='{ $paper}' limit 0,3");

```

```

4437 while ($tmp2_row=mysqli_fetch_assoc ($tmp2))
4448 $name_tmp=$name_tmp."<a href=/result_tmp_paper.php?tmp_id=" .
445     $tmp2_row[ 'paperid' ].">" . $tmp2_row[ 'title' ]."</a>" . "<br>";
4469 echo "<h>Title:{ $row[ 'title' ]}<h>";
4470 echo "</br></br>";
4481 echo "<p>Reconmmanded paper:{ $name_tmp}</p>";
4492 echo "</br></br>";
4503 echo "<p>Paperpublishyear:{ $row[ 'paperpublishyear' ]}</p>";
4514 echo "</br></br>";
4525 echo "<p>authorname:<a href=/author11.php?id=$row2[ authorid]>{
453     $row2[ 'authorname' ]} </a></p> ";
4546 mysqli_free_result ($query2);
4557 mysqli_free_result ($tmp2);
4568 $name_tmp='';
4579 }

```

458 2.2.2. part 2—beautify the web page

459 In this part, we will show how we beautified the website frontend using
460 HTML commands and css files.

461 *Home page.* Firstly, we downloaded a mature website template from make8.com.
462 It contained an HTML page and a css file. Then we made changes to the
463 original files.

464 Our beautification is mained done on the homepage. The whole homepage
465 is divided into two parts: the searching part which immediately shows up
466 when connecting the website, and the part that shows our group members'
467 names, which needs the users to roll down to the end of the page. In the
468 searching part, three searching boxes are used to search different messages.
469 Two background photos are added to the website, one can roll up and down
470 with the searching boxes, and the other is fixed as a background.

471 The css files allows us to set the specific scale of each part, and every part
472 are independent from each other. When calling the settings, it only needs a
473 class command to use them. By this way, the html commands and the logic
474 can be clearer, and the change of appearance and the change of parts are
475 independent from each other.

476 The details of our final works on beautification are shown as follows:
477 Title command allows the website open with a title, otherwise it will show
478 the default name of its root file.

```
4791 <title>Home</title>
```

480 An floating button is added on the fixed part of the page however the
481 page rolls.

```
482.1 <nav class="cd-stretchy-nav edit-content">  
483.2 <a class="cd-nav-trigger" href="#0"> Menu <span aria-hidden="true"></span> </a>  
484.3 <span aria-hidden="true" class="stretchy-nav-bg"></span>  
486.4 </nav>
```

487 The banner part completes the main functions of the homepage. We have
488 modified the original css file, thus the boxes and buttons with the effects we
489 want can be called in this class. For details, the information each box can
490 search, such as "Author", "Conference" and "Papertitle", can be shown in the
491 boxes using "placeholder=..." settings.

```
492.1 <div class="banner-w3ltext">  
493.2 <div class="container">  
494.3 <h2>SEARCHING!</h2>  
495.4 <h3>start here</h3>  
496.5 </div>  
497.6 <div class="contact-w3ls-row">  
498.7 <form action="result11.php" method="get">  
499.8 <input class="author" type="text" id="name" name="name"  
500.9 placeholder="Author">  
501.9 <div class="col-md-2 col-xs-4 place-grid">  
502.0 <input class="wthree-btn" type="submit" data-toggle="modal"  
503.1 value="Search"><br>  
504.1 </div>  
505.2 </form>  
506.3 <script>  
507.4 $(document).ready(function(){  
508.5 $('#name').autocomplete({  
509.6 source: "hint.php",  
510.7 minLength: 1  
511.8 });  
512.9 });  
513.0 </script>  
514.1 <p class="wthree-ttext"></p>  
515.2 <form action="result_conference2.php" method="get">  
516.3 <input class="conference" type="text" id="conference" name="conference" placeholder="Conference">  
517.4 <div class="col-md-2 col-xs-4 place-grid">  
518.5 <input class="wthree-btn" type="submit" data-toggle="modal"  
519.6 value="Search"><br>  
520.7 </div>  
521.8 </form>
```

```

5238 <script>
5249     $(document).ready(function() {
5250     $('#conference').autocomplete({
5261         source: "hint_2.php",
5272         minLength: 1
5283     });
5294 });
5305 </script>
5316 <p class="wthree-ttext"></p>
5327 <form action="result_paper1.php" method="get">
5338 <input class="paper" type="text" id="papertitle" name="
534     papertitle" placeholder="Papertitle">
5359 <div class="col-md-2 col-xs-4 place-grid">
5360 <input class="wthree-btn" type="submit" data-toggle="modal"
537     value="Search"><br>
5381 </div>
5392 </form>
5403 <script>
5414     $(document).ready(function() {
5425     $('#papertitle').autocomplete({
5436         source: "hint_3.php",
5447         minLength: 1
5458     });
5469 });
5470 </script>
5481 <p class="wthree-ttext"><br></p>
5492 </div>
5503 </div>

```

551 We didn't do much change to the slide effect. The codes are shown as
552 follows:

```

5531 <script defer src="js/jquery.flexslider.js"></script>
5542 <script type="text/javascript">
5553 $(window).load(function() {
5564     $('#flexslider').flexslider({
5575         animation: "slide",
5586         start: function(slider) {
5597             $('body').removeClass('loading');
5608         }
5619     });
5620 });
5631 </script>
5642 <script type="text/javascript">
5653     jQuery(document).ready(function($) {
5664         $(".scroll").click(function(event) {

```

```

5675     event.preventDefault();
5686     $('html,body').animate({scrollTop:$(this.hash).offset().top
569     },1000);
5707     });
5718     });
5729 </script>
5730 <script src="js/modernizr.js"></script> <!-- Modernizr -->
5741 <script src="js/menu.js"></script> <!-- Resource jQuery -->
5752 <script src="js/jquery.filterizr.js"></script>
5763 <script src="js/controls.js"></script>
5774 <script type="text/javascript">
5785     $(function() {
5796     $('.filtr-container').filterizr();
5807     });
5818 </script>
5829 <!-- swipe box js -->
5830 <script src="js/jquery.swipebox.min.js"></script>
5841 <script type="text/javascript">
5852     jQuery(function($) {
5863     $(".swipebox").swipebox();
5874     });
5885 </script>

```

589 *Other pages.* And in other pages, we set the background photos, the color
590 and size of words and the size of button. 1.set the background photo

```

5911 <head>
5922     <meta charset="utf-8">
5933     <meta name="viewport" content="width=device-width, initial-
594     scale=1">
5954     <link rel="stylesheet" href="https://apps.bding.com/libs/
596     bootstrap/3.2.0/css/bootstrap.min.css">
5975 </head>
5986 <style>
5997 #example1 {
6008     background-image: url(3.jpg);
6019     background-position: right bottom, left bottom;
6020     background-repeat: repeat, repeat;
6031     padding: 100px;
6042     background-size: 100% 100%;
6053 }
6064 </style>
6075 <body>
6086 <div id="example1">
6097 //other code

```



```
6108 </div>
6119 </body>
```

612 2.set the color and size of words

```
6131 <style>
6142 h {font-size:40px;}
6153 h{ color:#FF8DC}
6164 </style>
6175 <body>
6186 <center><h>Conference Name: <?php echo $_GET["conference"]; ?><
619     br></h></center>
6207 <center><h>The results are as follows:</h><br>
6218 </body>
```

622 3.set the size of button

```
6231 <center>
6242 <button id="button1" onclick="previous()" style="height:35px;
625     width:70px;">Previous</button>
6263 <button id="button2" onclick="next()" style="height:35px;width:70
627     px;">Next</button></center>
```

628 *2.2.3. part 3—the graph of the author relationship*

629 First, we have to find all people who has cooperated with the author and
630 judge whether he/she is the author's teacher or student using the classification
631 trained in lab three. To speed up the searching process, I choose to write all
632 the relationship into a new table so the program just needs to search in the
633 table while the user is operating. Second, make the relationship visualizable
634 by using a picture. This should be written into a .json file and the main
635 process has been given in the net provided.

636 Create New Table

637 First, find all the authors and their ID that have published the same
638 paper. The sql sentence is like the following shows.

```
6391 a=""select AuthorID,AuthorName from authors""
6402 b=""select PaperID from paper_author_affiliation
6413     where AuthorID='{ }'"".format(author1)
6424 c=""select AuthorID from paper_author_affiliation
6435     where PaperID='{ }'"".format(paper[0])
6446 d=""select AuthorName from authors
6457     where AuthorID='{ }'"".format(author2)
```

646 Then, we have to judge whether the second one is the teacher or student
647 of the first one. With the preparation of lab three, it's easy to execute. We

648 use label1 to show if the former is the teacher of the latter, and label2 to
 649 show if the former is the student of the latter. The code in python is like the
 650 following.

```
651 lr = LogisticRegression()
652 lr=joblib.load('lr.model')
653 feature1=feature(author1,author2,cursor)
654 feature2=feature(author2,author1,cursor)
655 test1=np.array(feature1)
656 test2=np.array(feature2)
657 test1=test1.reshape(1,-1)
658 test2=test2.reshape(1,-1)
659 label1=lr.predict(test1)
660 label2=lr.predict(test2)
```

661 Next, it's time to create a new table and insert data into it. The sql
 662 sentence to insert data is:

```
663 sql_="""insert into co_relation
664 (Author1,Name1,Author2,Name2,Label1,Label2)
665 values({0},{1},{2},{3},{4},{5});"""
666 .format(author1,name1[1],author2,name2[0][0],int(label1
667 [0]),int(label2[0]))
668 cursor.execute(sql_)
```

669 Write .json File

670 First, find all the related person and their ID and write them into .json
 671 file as nodes. The code is partly shown below:

```
672 $array = array();
673 $arrayid=array();
674 $label1=array();
675 $label2=array();
676
677 $sql1="SELECT Author2,Name2,Label1,Label2 FROM co_relation WHERE
678 Author1=" . $t . " ";
679 $result1 = mysqli_query($con,$sql1);
680 while($row1=mysqli_fetch_array($result1,MYSQLNUM))
681 {
682     array_push($arrayid,$row1[0]);
683     array_push($array,$row1[1]);
684     array_push($label1,$row1[2]);
685     array_push($label2,$row1[3]);
686 }
687
688 $sql2="SELECT AuthorName FROM author WHERE AuthorID=" . $t . "
689 ";
```

```

690:7 $result2 = mysqli_query($con,$sql2);
691:8 $row2=mysqli_fetch_array($result2,MYSQLNUM);
692:9
693:0 $len=sizeof($arrayid);
694:1
695:2 $filename = "D:/xampp_/htdocs/tmp.json";
696:3 $handle=fopen($filename,"w");
697:4 $content="{
698:5     \"nodes\": [
699:6     ";
700:7 $str=fwrite($handle,$content);
701:8 fclose($handle);
702:9
703:0 $handle=fopen($filename,"a+");
704:1 for($i=0;$i<$len;$i++)
705:2 {
706:3     if(((int)$label1[$i]==0 and (int)$label2[$i]==0) or (int)
707     $label1[$i]==1 and (int)$label2[$i]==1)
708:4     {
709:5         $content="        {\"id\": \"\".$array[$i].\" \".$arrayid[$i].\"
710     \",\"group\":4},
711:6     ";
712:7         $str=fwrite($handle,$content);
713:8     }
714:9 }
715:0
716:1 $content="    {\"id\": \"\".$row2[0].\" \".$t.\"\", \"group\":1}
717:2 ";
718:3 $str=fwrite($handle,$content);

```

719 Then, write their relationship into .json file as links(code also partly
720 showed).

```

721:1 for($i=0;$i<$len;$i++)
722:2 {
723:3     $content="        {\"source\": \"\".$row[0].\" \".$t.\"\", \"target
724     \": \"\".$array[$i].\" \".$arrayid[$i].\"\", \"value\":1},
725:4     ";
726:5     $str=fwrite($handle,$content);
727:6 }

```

728 Finally, we just need to use the tmp.json file to draw the picture.

729 3. Results Display

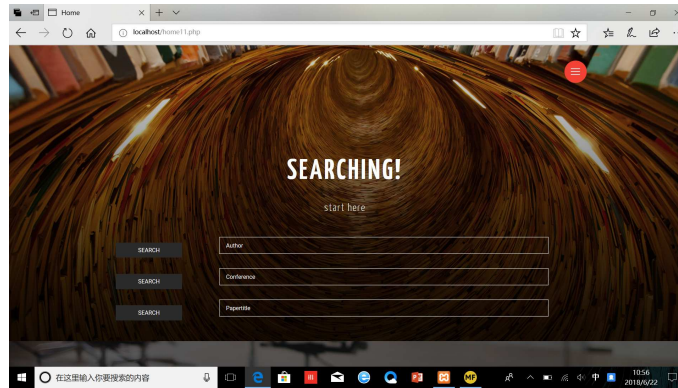


Figure 1: HOMESEARCH

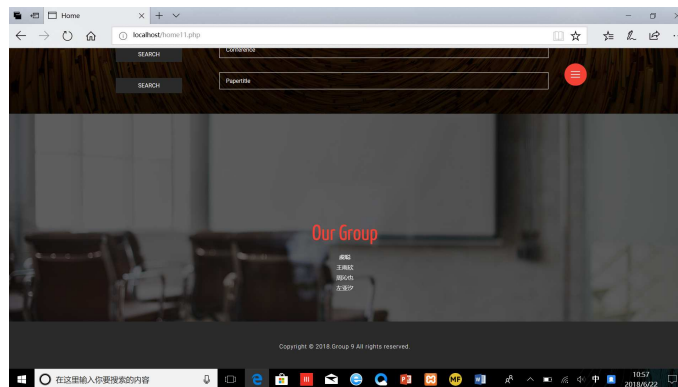


Figure 2: HOMESEARCH

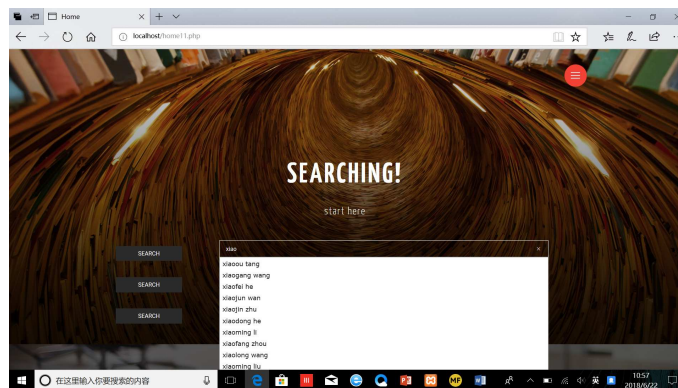


Figure 3: Automatic complementing

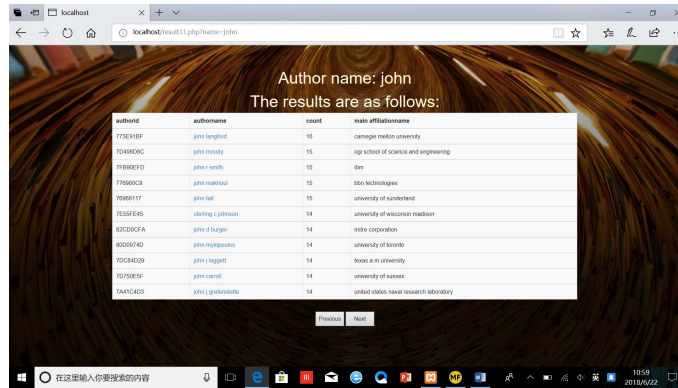


Figure 4: Result page

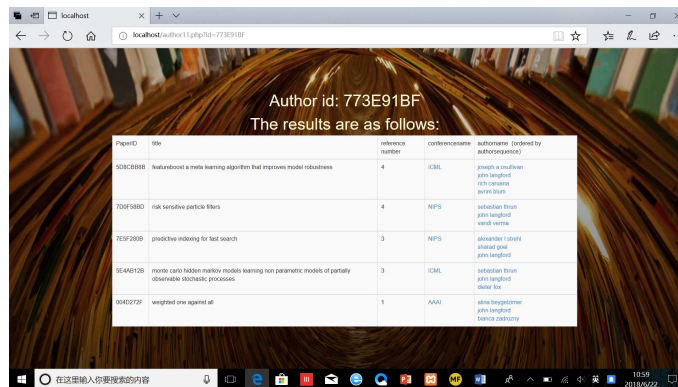


Figure 5: Author Page

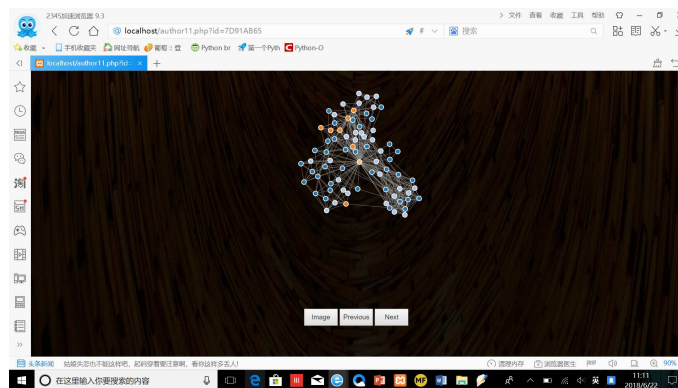


Figure 6: Image

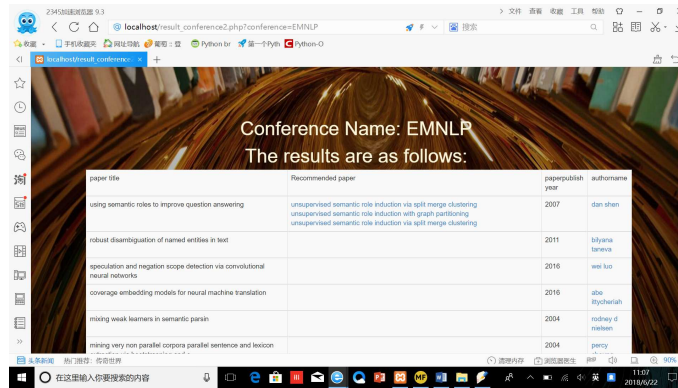


Figure 7: Conference Page

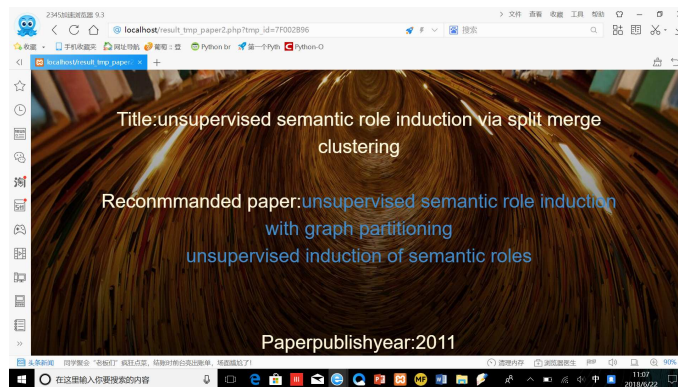


Figure 8: Refenrence Paper Page

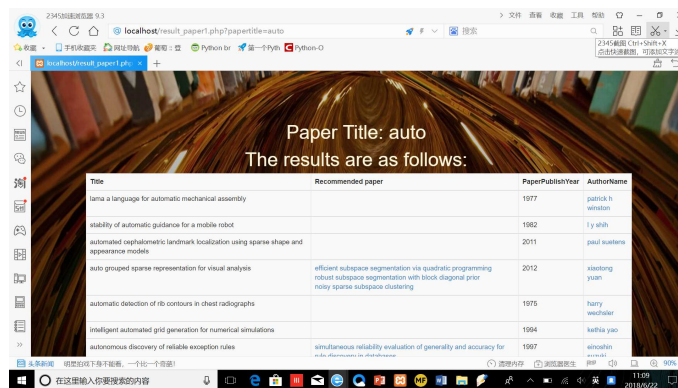


Figure 9: Paper Page