FINAL PROJECT

June 24, 2018

Authors:
517030910346 Liu Du
517030910316 Jiang Huangfei
517030910300 Yang Jinhai
517030910339 Fan Yu
Shanghai Jiao Tong University

Contents

0 Main Target and Mission Distribution

The main mission of our final project is to build a website with all of the knowledge we learn from this class.It's also suggested to stretch our mind and build something with which we have no contact before.

After discussion, the main mission distribution of our group is as below:

The basic expand of final website is done by Fanyu and JiangHuangfei,with fanyu building paper web page and doing paper recommendations and JiangHuangfei in charge of conference page.

YangJinhai is responsible for improvement of MYSQL sentences and database itself and learning to use elasticsearch to improve the speed of searching.

For the aspect of UI design and user interface, LiuDu take the lead.

The report and presentation of our work is mainly done by Fanyu and JiangHuangfei.Fanyu build the PPT and Jiang doing Latex and give introduction and summary in the report.

1 Paper Web Page & Conference Web Page & Paper recommendation

1.1 Problem Description

Paper Web Page: add the search of paper title into the search result page, and if there are more than 10 results, add a paper turning function into it.

Add thesis recommendation (such as recommendation through common authors, common references, etc.) The page is recommended for related papers, and can also be viewed on the scholar's page through this scholar's paper. Recommendations, etc.

Conference Web Page: build a website called conference.php with a search bar for users to enter conference name. And conference.php will show several conferences accordingly. Furthermore, build another website called conference_detail.php to offer more information on one exact conference.

1.2 Solution of Paper web page & paper recommendations

The search of paper was quiet similar to the search of author, we give the home page a title, and the home page give it back to TitleSearchFront.php, and this page select the information we need and print it on the website. If we need to turn a page, we post the title and the page into the TitleSearchBack.php, and this php also select the information out, and give it back to the TitleSearchFront.php, and replace the previous information in the website, then everything is done.

The recommendation of the paper is based on the number of the same authors between the recommended paper and the original paper. When we get a page of 10 papers, we want to select out the most similar paper to them. First we select out the

authors of every paper, store them in a list "SomeAuthor" ,and for every author in the list, we select out all papers he writes. And for each one of these papers, we select out all authors of this paper, so we compare these authors with the authors in the list "SomeAuthor", count how many authors are the same. And finally select out the most similar paper.

1.3 Source Code

```
var page = 1;
function PreviousPage(PaperTitle)
{
    page = page - 1;
    $.ajax
    ({
        url:"paper_back.php",
        type:'post',
        dataType:"json",
        data:{'PaperTitle':PaperTitle,'Page':page},
        success:function (data)
        {
            for(var i = 0;i < data.length; i++)
            {
                var row = data[i];
                var table = document.getElementById('Papers');
                var name = table.rows[i+1].cells[0];
                var id = table.rows[i+1].cells[1];
                var year = table.rows[i+1].cells[2];
                var aid = table.rows[i+1].cells[3];
                name.innerHTML = row[0];
                id.innerHTML = row[1];
                year.innerHTML = row[2];
                aid.innerHTML = row[3];
            }
            var row2 = data[data.length-1];
            var data2='<tr><td>'+row2[0]+'</td><td>'
            +row2[1]+'</td><td>'+row2[2]+'</td><td>'
            +row2[3]+'</td></tr>';
            $('#RecommondPaper').html(data2);

        }
    });
}

```

```
36
37 $sql = "SELECT Title,PaperID,PaperPublishYear,ConferenceID
38         FROM papers
39         WHERE Title LIKE '%$PaperTitle%'
40         GROUP BY PaperID
41         ORDER BY PaperPublishYear DESC LIMIT 0, 10";
42
43 echo '<h2>Papers<h2>';
44 echo '<table border="1" id = "Papers" align="center">
45     <tr><td>PaperTitle</td><td>PaperID</td>
46     <td>PaperPublishYear</td><td>ConferenceID</td></tr>';
47 $RescommondResult= array();$AuthorALL=array();$Recommond='';
48 $num = 0;$num2 = 0;$max=0;
49 while($row = mysqli_fetch_array($result))
50 {
51     echo '<tr><td> '.$row[0].' </td>
52         <td>'.$row[1]. '</td>
53         <td>'.$row[2]. '</td>
54         <td>'.$row[3].'</td>';
55     $SomeAuthor=array();
56     $PaperID=$row['PaperID'];
57     $sql2="SELECT AuthorID FROM
58     paper_author_affiliation WHERE PaperID='$PaperID'";
59     $result2 = mysqli_query($conn,$sql2);
60     while ($row2 = mysqli_fetch_array($result2))
61     {
62         array_push($SomeAuthor,$row2['AuthorID']);
63     }
64     $sql5="SELECT COUNT(AuthorID) AS Num
65      FROM paper_author_affiliation WHERE PaperID='$PaperID'";
66     $result5 = mysqli_query($conn,$sql5);
67     while($row5 = mysqli_fetch_array($result5)) {
68         $AuthorNum = $row5['Num'];
69     }
70     $AuthorALL[$num] = $SomeAuthor;
71     for($j=0;$j<$AuthorNum;$j++) {
72         $result3 = mysqli_query($conn,
73         " SELECT PaperID FROM paper_author_affiliation
74         WHERE AuthorID='$SomeAuthor[$j]'");
75
76         while ($row3 = mysqli_fetch_array($result3)) {
77             $PaperID3 = $row3['PaperID'];
78             $denote=0;
```

```
            if ($PaperID3 == $PaperID)
                continue;
            $result4 = mysqli_query($conn, " SELECT AuthorID
            FROM paper_author_affiliation WHERE PaperID='$PaperID3'");
            while ($row4 = mysqli_fetch_array($result4)) {
                for ($i = 0; $i < count($SomeAuthor); $i++)
                    if ($SomeAuthor[$i] == $row4["AuthorID"]
                     and $i!=$j)
                        $denote+=1;
            }
            if ($denote>$max)
                $max=$denote; $Recommond=$PaperID3;
        }
    }
    $num = $num + 1;
}
$sql6 = "SELECT Title,PaperID,PaperPublishYear,ConferenceID
        FROM papers
        WHERE PaperID = '$Recommond'";
$result6 = mysqli_query( $conn, $sql6 );
$row7=array();
while($row6 = mysqli_fetch_array($result6)){
    $row7 = $row6;
}
echo '</table>';
echo '<h2>RecommondPaper<h2>';
echo '<table border="1" align="center">??
    <thead>??
        <tr>
            <th>PaperTitle</th>
            <th>PaperID</th>
            <th>PaperPublishYear</th>
            <th>ConferenceID</th>
        </tr>??
    </thead>?
    <tbody id="RecommondPaper">
    </tbody> ? ?
    <tfoot>
    </tfoot>';
echo '<tr><td> '.$row7[0].' </td>
      <td>'.$row7[1]. '</td>
      <td>'.$row7[2]. '</td>
      <td>'.$row7[3].'</td>';
```

```
122 echo '</table>?';
123
124 echo "<button style=\"position:relative;left:500px\"
125  onclick=PreviousPage('{$PaperTitle}')>Previous</button>";
126 echo "<button style=\"margin-left:800px\"
127  onclick=NextPage('{$PaperTitle}')>Next</button>";
128 mysqli_close($conn);
129 ?>
130 </body>
131 </html>
```

```
1  <?php
2
3
4  paper_back.php
5
6
7  $PaperTitle = $_POST['PaperTitle'];
8  $Page = $_POST['Page'];
9  if($Page == 0)
10 {}
11 else
12 {
13     $conn = mysqli_connect('localhost', 'root', '');
14     if(! $conn )
15     {
16         die('Connection failed: ' . mysqli_error($conn));
17     }
18
19     mysqli_query($conn , "set names utf8");
20
21     $start = ($Page-1)*10;
22     $sql = "SELECT Title,PaperID,PaperPublishYear,ConferenceID
23         FROM papers
24         WHERE Title LIKE '%$PaperTitle%'
25         GROUP BY PaperID
26         ORDER BY PaperPublishYear
27         DESC LIMIT $start, 10";
28     mysqli_select_db( $conn, 'homework');
29
30
31     $result = mysqli_query( $conn, $sql );
32
```

```
33      if(! $result )
34      {
35          die('Could not get data: ' . mysqli_error($conn));
36      }
37
38      $result_all = array();
39      $RescommondResult= array();
40      $AuthorALL=array();$Recommond='';
41      $num = 0;$num2 = 0;$max=0;
42      while($row = mysqli_fetch_array($result))
43      {
44
45          $SomeAuthor=array();
46          $result_all[$num] = $row;
47          $PaperID=$row['PaperID'];
48          $sql2="SELECT AuthorID
49          FROM paper_author_affiliation
50          WHERE PaperID='$PaperID'";
51          $result2 = mysqli_query($conn,$sql2);
52          while ($row2 = mysqli_fetch_array($result2))
53          {
54              array_push($SomeAuthor,$row2['AuthorID']);
55          }
56          $sql5="SELECT COUNT(AuthorID) AS Num
57          FROM paper_author_affiliation
58          WHERE PaperID='$PaperID'";
59          $result5 = mysqli_query($conn,$sql5);
60          while($row5 = mysqli_fetch_array($result5)) {
61              $AuthorNum = $row5['Num'];
62          }
63          $AuthorALL[$num] = $SomeAuthor;
64          for($j=0;$j<$AuthorNum;$j++) {
65              $result3 = mysqli_query($conn, " SELECT PaperID
66              FROM paper_author_affiliation
67              WHERE AuthorID='$SomeAuthor[$j]'");
68              while ($row3 = mysqli_fetch_array($result3)) {
69                  $PaperID3 = $row3['PaperID'];
70                  $denote=0;
71                  if ($PaperID3 == $PaperID){
72                      continue;
73                  }
74                  $result4 = mysqli_query($conn, " SELECT AuthorID
75                  FROM paper_author_affiliation WHERE PaperID='$PaperID3'");
```

```
                while ($row4 = mysqli_fetch_array($result4)) {
                    for ($i = 0; $i < count($SomeAuthor); $i++) {
                        if ($SomeAuthor[$i] == $row4["AuthorID"]
                            and $i!=$j){
                            $denote+=1;
                        }
                    }
                }
                if ($denote>$max){
                    $max=$denote; $Recommond=$PaperID3;
                }
            }
        }
        $num = $num + 1;
    }
    $sql6 = "SELECT Title,PaperID,PaperPublishYear,ConferenceID
        FROM papers
        WHERE PaperID = '$Recommond'";
    $result6 = mysqli_query( $conn, $sql6 );
    while($row6 = mysqli_fetch_array($result6)){
        $result_all[$num] = $row6;
    }
    mysqli_close($conn);
    echo json_encode($result_all);
}
?>
```

1.4 Code Expanation

1.4.1 paper_front.php

We create two tables "Papers" and "RecommendPaper" to store the information of papers and the recommend paper.

```
echo '<table border="1" id = "Papers" align="center"><tr>
<td>PaperTitle</td><td>PaperID</td>
<td>PaperPublishYear</td><td>ConferenceID</td></tr>';

echo '<table border="1" align="center">??
    <thead>??
        <tr>
            <th>PaperTitle</th>
            <th>PaperID</th>
```

```
            <th>PaperPublishYear</th>
            <th>ConferenceID</th>
        </tr>??
    </thead>?
    <tbody id="RecommondPaper">
    </tbody> ? ?
    <tfoot>
    </tfoot>';
```

And then we bind events into the button of turning a page.

```
echo "<button style=\"position:relative;
left:500px\" onclick=PreviousPage('{$PaperTitle}')>Previous</button>";
echo "<button style=\"margin-left:800px\"
 onclick=NextPage('{$PaperTitle}')>Next</button>";
```

For paper_front.php, we create a function to post the information back into the back end, and receive the information from the back end, and parser the json information, and then replace the information in the website.

We use the function of the previous page as an example.

```
function PreviousPage(PaperTitle)
{
    page = page - 1;
    $.ajax
    ({
        url:"paper_back.php",
        type:'post',
        dataType:"json",
        data:{'PaperTitle':PaperTitle,'Page':page},
        success:function (data)
        {
            for(var i = 0;i < data.length; i++)
            {
                var row = data[i];
                var table = document.getElementById('Papers');
                var name = table.rows[i+1].cells[0];
                var id = table.rows[i+1].cells[1];
                var year = table.rows[i+1].cells[2];
                var aid = table.rows[i+1].cells[3];
                name.innerHTML = row[0];
                id.innerHTML = row[1];
                year.innerHTML = row[2];
                aid.innerHTML = row[3];
```

```
            }
            var row2 = data[data.length-1];
            var data2='<tr><td>'+row2[0]+'</td><td>'
                +row2[1]+'</td><td>'+row2[2]+'</td>
                <td>'+row2[3]+'</td></tr>';

        }
    });
}
```

Notice: we add the recommend paper into the last piece of the information, so the variable "data2" means the recommend paper. And we print the recommend paper in another table.

Because when we turn from the home.php to the paper_front.php, we don't need to click the button in the paper_front.php,so we don't post information to the paper_back.php, and don't have information return. So in the paper_front.php, we need to select the first ten pieces of information and print them in the website. This is just what we do in the paper_back.php, and I will just give explanation of the paper_back.php, and the programm in the paper_front.php is the same principle.

1.4.2 paper_back.php

First, get post information and connect the database

```
paper_back.php
$PaperTitle = $_POST['PaperTitle'];
$Page = $_POST['Page'];
if($Page == 0)
{}
else
{
    $conn = mysqli_connect('localhost', 'root', '');
    if(! $conn )
    {
        die('Connection failed: ' . mysqli_error($conn));
    }

    mysqli_query($conn , "set names utf8");

    $start = ($Page-1)*10;
```

Then, select information of the papers which have the similar names of the input title. And create some array to store information.

```
$sql = "SELECT Title,PaperID,PaperPublishYear,ConferenceID
    FROM papers
    WHERE Title LIKE '%$PaperTitle%'
    GROUP BY PaperID
    ORDER BY PaperPublishYear DESC LIMIT $start, 10";
mysqli_select_db( $conn, 'homework');


$result = mysqli_query( $conn, $sql );

if(! $result )
{
    die('Could not get data: ' . mysqli_error($conn));
}
$result_all = array();
$RescommondResult= array();
$AuthorALL=array();
$Recommond='';
$num = 0;$num2 = 0;$max=0;
```

For the ten papers selected out, we select all authors ID and the number of authors of each paper, and store them in the array "SomeAuthor" and an variable "AuthorNum".

```
while($row = mysqli_fetch_array($result))
    {

    $SomeAuthor=array();
    $result_all[$num] = $row;
    $PaperID=$row['PaperID'];
    $sql2="SELECT AuthorID
    FROM paper_author_affiliation
    WHERE PaperID='$PaperID'";
    $result2 = mysqli_query($conn,$sql2);
    while ($row2 = mysqli_fetch_array($result2))
    {
        array_push($SomeAuthor,$row2['AuthorID']);
    }
    $sql5="SELECT COUNT(AuthorID) AS Num
     FROM paper_author_affiliation WHERE PaperID='$PaperID'";
    $result5 = mysqli_query($conn,$sql5);
    while($row5 = mysqli_fetch_array($result5)) {
        $AuthorNum = $row5['Num'];
    }
```

For every author in the array "SomeAuthor", we select all papers he wrote, and all authors' ID of his papers

```
for($j=0;$j<$AuthorNum;$j++) {
$result3 = mysqli_query($conn, " SELECT PaperID
    FROM paper_author_affiliation
    WHERE AuthorID='$SomeAuthor[$j]'");
while ($row3 = mysqli_fetch_array($result3)) {
    $PaperID3 = $row3['PaperID'];
    $denote=0;
    if ($PaperID3 == $PaperID) {
    continue;
}
$result4 = mysqli_query($conn, " SELECT AuthorID
    FROM paper_author_affiliation
    WHERE PaperID='$PaperID3'");
```

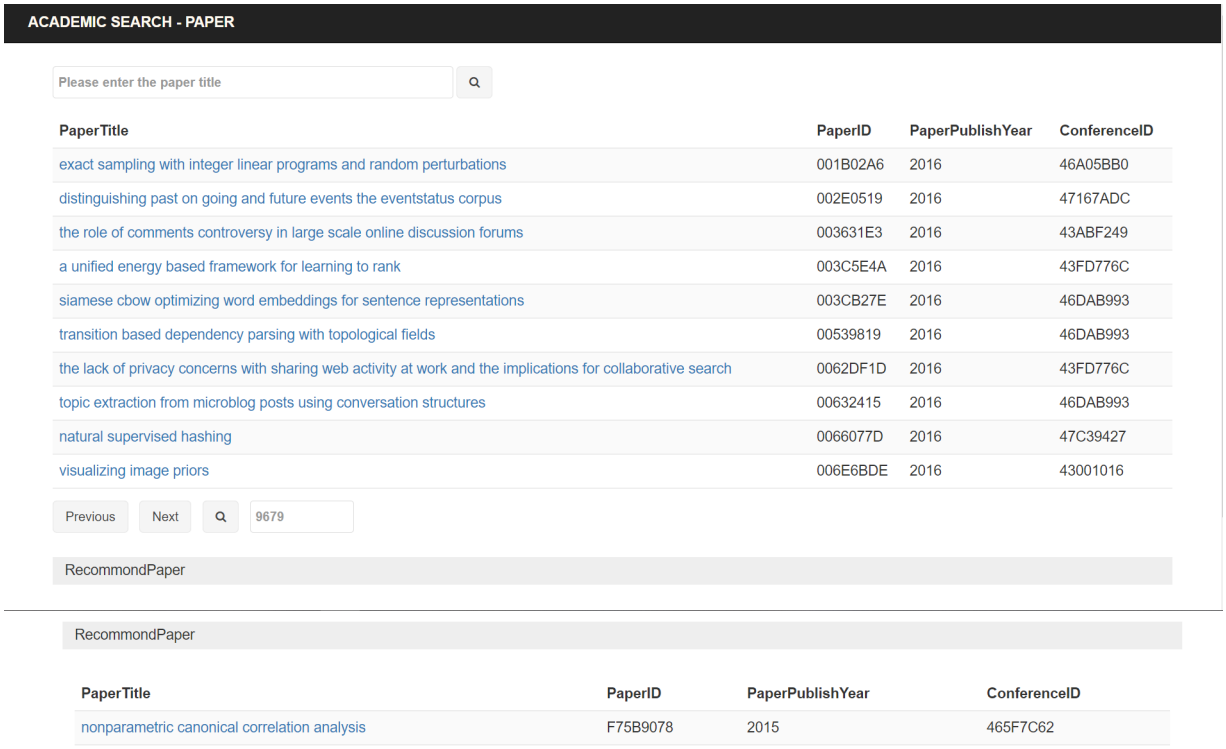
We compare these authors with the authors in array"SomaAuthor", and pick out the paper which have the most common authors.

```
while ($row4 = mysqli_fetch_array($result4)) {
    for ($i = 0; $i < count($SomeAuthor); $i++) {
    if ($SomeAuthor[$i] == $row4["AuthorID"] and $i!=$j){
        $denote+=1;
        }
    }
}
if ($denote>$max){
    $max=$denote; $Recommond=$PaperID3;
}
```

And the recommended paper into the last position of the array including all information, "result_all". And return these data into paper_front.php and finally display them.

```
$sql6 = "SELECT Title,PaperID,PaperPublishYear,ConferenceID
    FROM papers
    WHERE PaperID = '$Recommond'";
$result6 = mysqli_query( $conn, $sql6 );
while($row6 = mysqli_fetch_array($result6)){
    $result_all[$num] = $row6;
}
mysqli_close($conn);
echo json_encode($result_all);
```

1.5 Result Display



1.6 Problems turned out

The first problem is that where we should store the recommend paper. At first I think I should use one array to store the results of paper search, one array to store the recommend paper. But then I find it difficult to deal with two json array in the paper_front.php. Finally, I realized it will be quiet easy if I just add the recommend paper in the last position of the array of search result.

When I run my programm, I find it takes a long time. I check it for serval times and I find because I only use serval columns in the labs before, so I only create index on some certain columns and forget to create indexes on other columns. I need to be careful next time.

When I parser json information int paper_front.php, there was always an error "Invalid charaset at positon 1".I found because in paper_back.php, I echo other things besides the information we need,and these things were not in json form,so we can not parser it.After I delete the echo sentences,it made sense.

When I finished the paper_front.php, I found my button did not work,after checking it a longtime,I found that I put that button at the end of the html,while used it before I defined it.So when the event was triggered, the button did not exist.At last I put the button before the JS code and it made sense.

1.7 Solution of Conference page

1.7.1 conference search

The search of conference was quiet similar to the search of paper.Since my teammate FanYu has delivered his method in this report. I will not offer the similiar part in my part.After setting POST deliver method,and build conference.php accordingly. And below is the SQL codes, which is seen as the main difference to paper search.

```
$sql = "select conferences.Conferencename, conferences.ConferenceID,
    count(papers.PaperID) as times
from conferences
inner join papers
on conferences.ConferenceID = papers.ConferenceID
where  Conferences.Conferencename like '%$ConferenceName%'
group by Conferences.Conferencename
order by times desc limit 10
";
```

1.7.2 Build conference detail

I started to build conference_detail.php. Once talking about one conference, we will think immediately about papers launched on it.So I get paper information according to conferenceID,and take orders based on the citation times , which is stored in paper reference table.SQL codes is as below:

```
$sql = "select papers.Title ,papers.PaperID,
    count(paper_reference.referenceID)
from papers
join paper_reference
on papers.paperID=paper_reference.referenceID
where ConferenceID='$ConferenceID'
group by papers.paperID
order by count(paper_reference.referenceID) desc
limit 0,10";
```

1.7.3 Turning Page

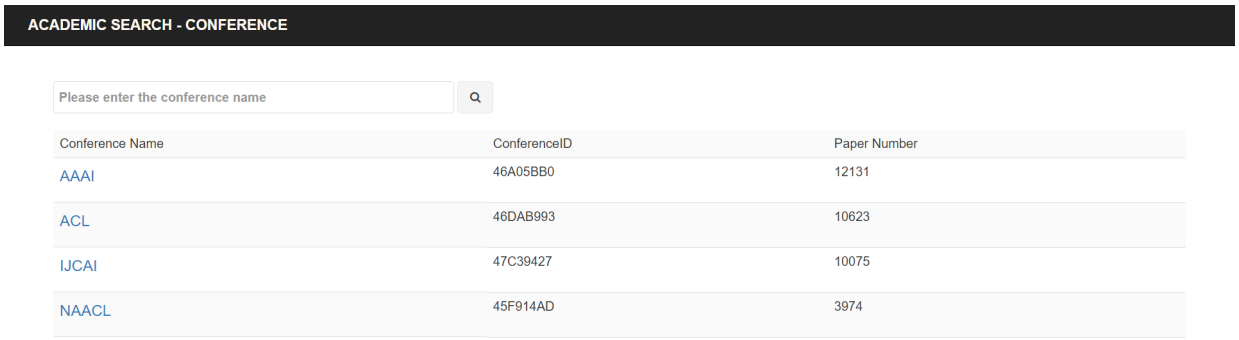
It's also necessary to add a function to turn pages. With posting conferenceID back into the back end and a series of operations, corresponding information will be replaced in the website.Codes of conference_detail_hint is shown as below(conference_hint is actually the same.)

```
<?php
```

```
$ConferenceID = $_POST['ConferenceID'];
$Page= $_POST['Page'];
if ($Page==0)
{}
else
{
    $conn = mysqli_connect('localhost', 'root', '');

  mysqli_query($conn , "set names utf8");

  $start= ($Page-1)*10;
  $sql = "select papers.Title ,papers.PaperID,
      count(paper_reference.referenceID)
from papers
join paper_reference
on papers.paperID=paper_reference.referenceID
where ConferenceID='$ConferenceID'
group by papers.paperID
order by count(paper_reference.referenceID) desc
limit $start, 10 "
;

    mysqli_select_db( $conn, 'papersearch');

    $result = mysqli_query( $conn, $sql );

    if(! $result )
    {
        die('Could not get data: ' . mysqli_error($conn));
    }

    $result_all = array();
  $num=0;
while ($row= mysqli_fetch_array($result))
  {
  $result_all[$num] = $row;

  $ConferenceID= $row[1];

  echo '<tr><td> <p><a
      href="conference_detail.php?ConferenceID='.$row[1].'">'.$row[0].'</a></p>
      </td>
  <td>'.$row[1].'</td>
```

```
42    <td>'.$row[2].'</td>';
43
44    $num = $num + 1;
45    }
46    mysqli_close($conn);
47      echo json_encode($result_all);
48
49  }
50
51  ?>
```

1.8 Result Display



2 SQL IMPROVEMENT

2.1 Problem Description

To improve the speed of searching, the speed of searching in database should be improved. Therefore, two ways can be considered: one is to improve the MySQL sentence, the other is to improve the database.

2.2 Solution Design

At first, we consider improving the MySQL sentence to raise the efficiency of searching in database. However, the sentences we design are not so complex and the grammar we use such as 'join', 'group by' and 'order' can not be replaced or deleted. Therefore,

there leave only one way for us to improve the speed of searching: to improve the table in the database.

If we use the table established before, when we search for something, we need to join two or even more table together to search for the exact information we want. Joining tables together causes a lot of time. So, the thought of combining the tables together to avoid joining tables came to our mind. We plan to establish some tables to combine the information together according to the information we need within a search. So, we create three new tables which are designed for three websites.

The first table is designed for the 'result' page. We select AuthorName, AuthorID, PaperNumber, AffiliationID, AffiliationName to form a new table serving for the result page.

We select PaperID, Title, PaperPublishYear, Citation, ConferenceID, ConferenceName to form another table serving for the author page.

And we select PaperID, Title, PaperPublishYear, Citation, ConferenceID, ConferenceName, AuthorNumber, AuthorName, AuthorID and AuthorSequence to form a new table to help us show the detail of an author.

Then the MySQL sentence can be simplified like this:

```
1 SELECT AuthorName, AuthorID, PaperNumber, AffiliationID, AffiliationName
    FROM authors
```

3 ELASTICSEARCH

3.1 Problem Description

The Elasticsearch is a more efficient tool for searching and we would like to use it to replace the MySQL. At first there exist two choices for us, one is Elasticsearch, the other is Solr. So, we search on the Google to compare their difference. There are two important aspects for us to consider. The first is the searching speed. The second is whether it's user-friendly.

The first aspect is about the searching speed. Elasticsearch is a newly technology and it performs better at updating data and searching for rapidly changing data because of per-segment caches. But Solr does better in static data because of caches and uninverted reader which is more suitable for our need of searching.

Next is about whether it's user-friendly. Elasticsearch is very user-friendly and it has very good APIs but Solr is not as good as Elasticsearch. However, considering for the current situation that Elasticsearch is becoming more and more popular as Stack Overflow, Amazon, eBay and Wikipedia are using it to help manipulate the data, we decide to use Elasticsearch to search.

3.2 Solution Design

At first, we need to import data from the original files given in the first lab and the database we created to create the index in Elasticsearch which is equivalent to the table in MySQL database. When we tried to learn how to use Elasticsearch, we found out that it's compiled by Java and we have no idea with it. Therefore, we decided to use the API programmed by other programmers. Comparing different ways on the internet (Python, PHP and etc.), we find that Elasticsearch has a good API (Application Programming Interface) with Python.

So, at first, we used the Python to create index in Elasticsearch. The first step we take is to write the data into a txt file using the previous files and the tables in database. The code is written as below:

```
import pymysql
db = pymysql.connect("localhost", "root", "", "main_db", charset="utf8")

cursor = db.cursor()

f1 = open(r"C:\Users\14838\Desktop\papers_es.txt", 'w+', encoding="utf8")
f1.truncate()

f = open(r"C:\Users\14838\Desktop\relations.txt", encoding='utf-8')

for line in f:
    line = line.rstrip('\n')
    if line == str("INSERT INTO `papers_all` (`PaperID`, `Title`,
        `PaperPublishYear`, `ConferenceID`, `cite`) VALUES"):
        continue
    else:
        lst = line.strip(',();')
        lst = lst.split(", ")
        PaperID = lst[0].strip("'")
        Title = lst[1].strip("'")
        PaperPublishYear = lst[2]
        ConferenceID = lst[3].strip("'")
        Citation = lst[4]

        author = """SELECT AuthorID, AuthorSequence
                    FROM paper_author_affiliation
                    WHERE PaperID = '{}'""".format(PaperID)
        cursor.execute(author)
        authors = cursor.fetchall()

        authorNum = """SELECT COUNT(AuthorID)
```

```
                        FROM paper_author_affiliation
                        WHERE PaperID = '{}'""".format(PaperID)
        cursor.execute(authorNum)
        author_num = cursor.fetchall()
        AuthorsNum = author_num[0][0]

        conference = """SELECT ConferenceName
                       FROM Conferences
                       WHERE ConferenceID = '{}'""".format(ConferenceID)
        cursor.execute(conference)
        conference_ = cursor.fetchall()
        if len(conference_) == 0:
            continue
        else:
            ConferenceName = conference_[0][0]

        for j in range(len(authors)):
            AuthorID = authors[j][0]
            AuthorSequence = authors[j][1]
            author_name = """SELECT AuthorName
                            FROM authors
                            WHERE AuthorID = '{}'""".format(AuthorID)
            cursor.execute(author_name)
            authors_name = cursor.fetchall()
            for k in range(len(authors_name)):
                AuthorName = authors_name[k][0]
                f1.write(PaperID + '\t' + Title + '\t' +
                    str(PaperPublishYear) + '\t' + str(Citation) + '\t' +
                         ConferenceID + '\t' + ConferenceName + '\t'
                         + str(AuthorsNum) + '\t' + AuthorID + '\t' +
                             AuthorName + '\t' + str(AuthorSequence) + '\n')


f.close()
f1.close()
db.close()
```

Then we install the elasticsearch which provides common ground for all Elasticsearch-related code in Python:

```
pip install elasticsearch
```

Notice that the version of the Elasticsearch must be over 5.0. Therefore, we should check the version of the Elasticsearch before you install the API. Then we use the helpers

in the elasticsearch to insert the data into the index. It can translate the basic Python data types to json which means we can also use PHP to insert the data into the index. To raise the speed of inserting data, we can form a list and put all the data into it to insert the data in a time.

```
from elasticsearch import Elasticsearch
from elasticsearch import helpers


es = Elasticsearch()

f = open(r"C:\Users\14838\Desktop\authors_es.txt", encoding='utf-8')

actions = list()

i = 1

for line in f:
    lst = line.rstrip('\n')
    lt = lst.split("\t")
    action = {
        "_index": "author_all",
        "_type": "authors_information",
        "_id": i,
        "_source": {
            "AuthorID": lt[0],
            "AuthorName": lt[1],
            "PaperNum": int(lt[2]),
            "AffiliationID": lt[3],
            "AffiliationName": lt[4]
        }
    }
    i += 1
    actions.append(action)

if len(actions) > 0:
    helpers.bulk(es, actions)
```

After we created the index, we focused on how to use it to improve the speed of searching. For the convenience of using it, we install the head to manage the Elasticsearch and when we want to use it, we need to start it in the cmd opened in the folder named head (grunt server). Then we can follow the instruction on the internet to create the environment for using the Elasticsearch in PHP. Before using it, we should import

the autoload file:

```
require_once('vendor/autoload.php');
use Elasticsearch\ClientBuilder;
```

Then we can connect to the Elasticsearch and use the query to search the data we want in the index we created in the Elasticsearch:

```
  $hosts = ["localhost:9200"];
    $start = ($page - 1) * 10;

    $client = ClientBuilder::create()->setHosts($hosts)->build();

    $query = ["index" => "author_all",
        "type" => "authors_information",
        "body" => [
            "query" => ["match" => ["AuthorName" => $AuthorName]],
            "sort" => ["PaperNum" => ["order" => "desc"]],
            "size" => ($start + 10)
        ]
    ];
```

Then we will get a returned data written by json. We take out the data we need and encode it into a json file:

```
$resp = $client->search($query);
    $total_num = $resp["hits"]["total"];

    $result_all = array();
    for($i = 0; $i < min(10, $total_num-$start);$i++)
    {
        $index = $resp["hits"]["hits"][$start+$i]["_source"];

        $result_all[$i]["AuthorName"] = $index["AuthorName"];
        $result_all[$i]["AuthorID"] = $index["AuthorID"];
        $result_all[$i]["PaperNum"] = $index["PaperNum"];
        $result_all[$i]["AffiliationID"] = $index["AffiliationID"];
        $result_all[$i]["AffiliationName"] = $index["AffiliationName"];
    }
    echo json_encode($result_all);
```

4 UI Design & User Interface

4.1 Design Philosophy

4.1.1 Efficient

A product without high efficiency is of no value. So what I want to pursue is to make the project built on the expectations of my users, which means there won't be any trumpery design in it.

4.1.2 Concise

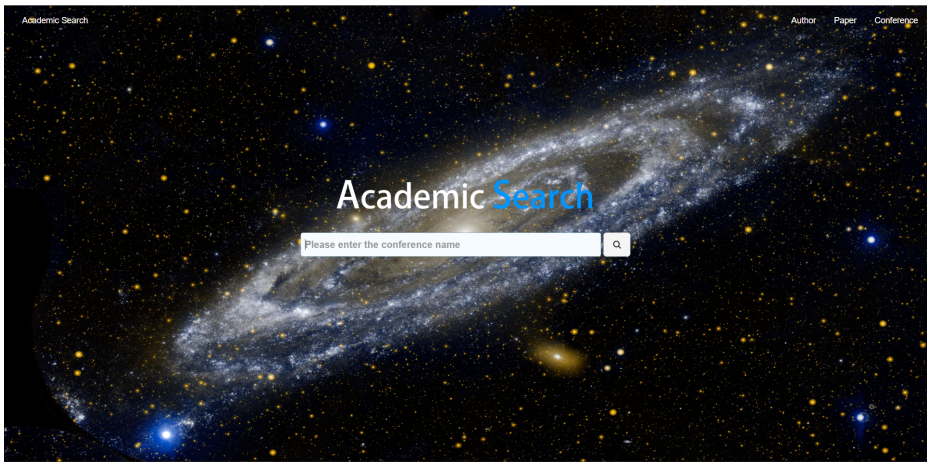
In this project, I avoid those ornate decorations like live wallpapers and turn to flat and concise design, for I position our project as the productivity tool and those decorations seem to be willing to supersede what really counts.

4.2 Front-end Framework

This project is mainly built on UIKIT. Besides, I also use JQueryUI and Bootstrap.

4.3 Home Page

4.3.1 The Overall Design



The home page is just like other mainstream search engine. To switch the object to search, just click the button on the right side of the navigation bar. For example, when you click the 'Paper' button the placeholder will be changed into 'Please enter the paper title' which means the object to search has been changed into paper.

4.3.2 Switcher

The realization of the switcher is based on UIKIT. In this part, I establish a one-to-one relationship between the switcher on the navigation bar and the input box. The code is as below.

```
<!-- only part of codes have showed-->

<nav id="mainNav" class="navbar uk-navbar-attached navbar-fixed-top">
    <div class="container-fluid">
        <div class="collapse navbar-collapse"
            id="bs-example-navbar-collapse-1">
            <ul class="nav navbar-nav
                navbar-right"data-uk-switcher="{connect:'#tb'}">

                <li><a href="" style="color: floralwhite">Author</a></li>
                <li><a href="" style="color: floralwhite">Paper</a></li>
                <li><a href="" style="color: floralwhite">Conference</a></li>
            </ul>
        </div>
    </div>
</nav>

<ul id="tb" class="uk-switcher">
    <li style="text-align: center">
        <form class="uk-form" action = "result.php" method = "get">
            <br>
            <label><input type = "text" name = "AuthorName" id= "key1"
                class="uk-form-large uk-form-width-large"
                placeholder="Please enter the author name"></label>
            <button type="submit" class="uk-button uk-button
                uk-button-large uk-button fa fa-search" ></button>
        </form>
    </li>
    <li style="text-align: center">
        <form class="uk-form" action = "paper_front_ver2.php" method =
            "get">
            <br>
            <label><input type = "text" name = "PaperTitle" id="key2"
                class="uk-form-large uk-form-width-large"
                placeholder="Please enter the paper title"></label>
            <button type="submit" class="uk-button uk-button
                uk-button-large uk-button fa fa-search" ></button>
        </form>
    </li>
    <li style="text-align: center">
        <form class="uk-form" action = "conference.php" method = "get">
            <br>
```

```
34            <label><input type = "text" name = "ConferenceName" id="key3"
                  class="uk-form-large uk-form-width-large"
                  placeholder="Please enter the conference name"></label>
35            <button type="submit" class="uk-button uk-button
                  uk-button-large uk-button fa fa-search" ></button>
36        </form>
37    </li>
38 </ul>
```

4.4 Result Page

4.4.1 Brief Information of the Result

When the user types the key word into the input box and jump to the result page, he will see a message box on the top of the website which will show the number of results. (This function is based on the 'Notify' component in UIKIT) The code is as below

```
1  <?php
2     //calculate the number of results
3     $AuthorName = $_GET['AuthorName'];
4     $conn = mysqli_connect('localhost', 'root', '',"main_db");
5     if(! $conn )
6     {
7         die('Connection failed: ' . mysqli_error($conn));
8     }
9     $sql = "SELECT COUNT(AuthorName) as num
10        FROM authors
11        WHERE AuthorName LIKE '%$AuthorName%'
12        ORDER BY num DESC";
13    $result=mysqli_query($conn,$sql);
14    $row=mysqli_fetch_row($result);
15    $sum=$row[0];
16
17    //.............................
18
19    //show the message box
20    echo '<script>UIkit.notify(" '.$sum;
21    echo " authors have been found.\");</script>";
22 ?>
```

4.4.2 Table Establishment

In each page, there will be 10 items shown. And each item includes author name, author id, paper number, affiliation id and affiliation name. Here I also use the table in UIKIT to lay out. The code is as below.

```
<?php
//the variable is the data source
echo "<div style='margin:1% 4% 0% 4%'>";
echo '<table class="uk-table uk-table-hover uk-table-striped"
    id="Result"><thead><tr><th>AuthorName</th><th>AuthorID\
</th><th>Papernumber</th><th>AffiliationID</th><th>\
AffiliationName</th></tr></thead>';
//uk-table-hover can highlight the item when mouse hovers on it
echo '<tbody>';
while($row = mysqli_fetch_array($result))
{
    $AuthorID = $row[1];
    echo '<tr><td> <p><a href = "author.php?AuthorID=\
    '.$row[1].'">'. $row[0]. '</a></p> </td>
        <td>'.$row[1]. '</td>
        <td>'. $row[2]. '</td>';
    $sql1 = "SELECT affiliationID, COUNT(*)
             FROM paper_author_affiliation
             WHERE AuthorID = '$AuthorID'
             GROUP BY affiliationID
             ORDER BY COUNT(*) DESC LIMIT 0, 1";
    $result1 = mysqli_query( $conn, $sql1 );
    if(! $result1 )
    {
        die('Could not get data: ' . mysqli_error($conn));
    }
    $row1 = mysqli_fetch_array($result1);
    if ($row1[0] == "None")
        echo '<td>None</td><td>None</td></tr>';
    else
    {
        $sql2 = "SELECT affiliationName
             FROM affiliations
             WHERE AffiliationID = '{$row1[0]}'
             ORDER BY COUNT(*)";
        $result2 = mysqli_query($conn, $sql2);
        if (!$result2) {
            die('Could not get data: ' . mysqli_error($conn));
```

```
        }
        $row2 = mysqli_fetch_array($result2);
        echo '<td>' . $row1[0] . '</td><td>' . $row2[0] . '</td></tr>';
    }

}
echo '</tbody></table></div>';
?>
```

4.4.3 Page Turning System

4.4.4 Previous button and Next button

Here I use ajax to request data from back-end. As I lay out the item in the form of table. It's not difficult to replace the current table with the new one. And I set a variable called to record the page number so that the page turning button can work. The code is as below.

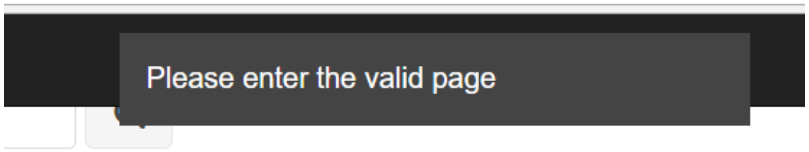
```
<script>
var page = 1;
function PreviousPage(AuthorName)
{
    page = page - 1;
    $.ajax
    ({
        url:"result_hint.php",
        type:'post',
        dataType:"json",
        data:{'AuthorName':AuthorName,'Page':page},
        success:function (data)
        {
            for(var i = 0;i < data.length; i++)
            {
                var row = data[i];
                var table = document.getElementById('Result');
                var name = table.rows[i+1].cells[0];
                var id = table.rows[i+1].cells[1];
                var num = table.rows[i+1].cells[2];
                var aid = table.rows[i+1].cells[3];
                var aname = table.rows[i+1].cells[4];
                name.innerHTML = "<a
                    href='author.php?AuthorID="+row[1]+"'>"+row[0]+'</a>';
                id.innerHTML = row[1];
                num.innerHTML = row[2];
```

```
                aid.innerHTML = row[3];
                aname.innerHTML = row[4];
            }
        }
    })
};
</script>
```

4.4.5 Jump to the given page

Apart from conventional page turning button, I add a text box that allow users to type in the page he want to jump to. As I have count the number of results in 4.1, here I set the place holder as the number of results divided by 10(sum/10+1) to show the total number of pages. Then I design a button to realize the function (Here I find that the type of the button should be set as 'button', otherwise, it will jump to a new page like 'localhost/result.php?page=44', which exactly not what I want.)

4.4.6 Invalid Page Number Input



Considering the invalid page number input, I set a validation check to make sure that the number I type in is valid. If it's invalid, a message box will pop up to alert users. The code is as below.

```
<script>
  function TypicalPage(AuthorName)
    {
        var sum= parseInt(<?php echo $sum;?>);
        sum=sum/10+1
        var page_copy = parseInt(document.getElementById("page").value);
        //get the page number from the text box
        if(page_copy>sum)
        {
            UIkit.notify("Please enter the valid page");
            return;
        }
        page=page_copy;
        $.ajax
        ({
            url:"result_hint.php",
```

```
            type:'post',
            dataType:"json",
            data:{'AuthorName':AuthorName,'Page':page},
            success:function (data)
            {
                for(var i = 0;i < data.length; i++)
                {
                    //var pagination=document.getElementById("pagination");
                    var row = data[i];
                    var table = document.getElementById('Result');
                    var name = table.rows[i+1].cells[0];
                    var id = table.rows[i+1].cells[1];
                    var num = table.rows[i+1].cells[2];
                    var aid = table.rows[i+1].cells[3];
                    var aname = table.rows[i+1].cells[4];
                    name.innerHTML = "<a
                        href='author.php?AuthorID="+row[1]+"'>"+row[0]+'</a>';
                    id.innerHTML = row[1];
                    num.innerHTML = row[2];
                    aid.innerHTML = row[3];
                    aname.innerHTML = row[4];

                }
            }
        })
    }
</script>


<?php
echo "<button type='button' class='uk-button uk-button uk-button-large
    uk-button fa fa-search'
    onclick=TypicalPage('{$AuthorName}')></button><span> </span>";
echo "<label><input type = \"text\" name = \"page\" id=\"page\"
    class=\"uk-form-large uk-form-width-mini\"
    placeholder=\"".$sum."\"></label>";
?>
```

4.5 Author Page

4.5.1 Cooperator Graph

As I have finished the process of drawing the graph in the Exp.4, so here I focus on how to show the graph properly. Here I use the component called 'off-canvas' in UIKIT. When I click the Graph button, there will be an off-canvas sidebar that slides in and out of the page. The code is as below.

```
<div id="my-id" class="uk-offcanvas">
    <div class="uk-offcanvas-bar">

        <!-- drawing the graph here -->

    </div>
</div>
```

4.5.2 Scollspy

In the author page. I put each item into a card and add a scrollspy to it. The Scrollspy component listens to page scrolling and triggers events based on the scroll position. For example, if I scroll down a page and an element appears in the viewport for the first time, I can trigger a smooth animation to fade in the element. The code is as below.

```
<?php
    echo "<div style='margin:auto 7% auto'><h2>AuthorName: {$AuthorName}
        <button class=\"uk-button uk-button-large\"
        data-uk-offcanvas=\"{target:'#graph'}\">Cooperator
        Graph</button><h2></div>";
    echo ' <div class="uk-grid" data-uk-grid-margin
        data-uk-scrollspy="{cls:\'uk-animation-fade uk-invisible\',
        target:\'> div > .uk-panel\', delay:300, repeat: true}">';
    $num=1;
    while($row1 = mysqli_fetch_array($result1))
    {
        echo '<div class="uk-width-medium-5-6" style="font-size:
            12px;margin: 1% 7% auto"><div class="uk-panel uk-panel-box
            uk-invisible"><p id="1'.$num.'" style="font-size:
            16px;font-weight: bolder"><a
            href="http://localhost/uikit/paper_front_ver2.php?PaperTitle\
        ='.$row1[0].'">'.$row1[0].'</a></p><dd><p
            id="2'.$num.'">Citation:'.$row1[2].'</p></dd>';
        $sql2 = "SELECT ConferenceName
              FROM conferences
```

```
            WHERE ConferenceID = '$row1[3]'
            ";
        $result2 = mysqli_query( $conn, $sql2 );

        if(! $result2 )
        {
            die('Could not get data: ' . mysqli_error($conn));
        }
        $row2 = mysqli_fetch_array($result2);


       echo '<dd><p id="3'.$num.'">Conference:'.$row2[0].'</p></dd><dd>';


        $sql3 = "SELECT AuthorID
            FROM paper_author_affiliation
            WHERE PaperID = '$row1[1]'
            ORDER BY AuthorSequence ASC";
        $result3 = mysqli_query( $conn, $sql3 );


        if(!$result3)
        {
            die('Could not get data: ' . mysqli_error($conn));
        }
        echo "<p id='4".$num."'>AuthorID:";
        while($row3 = mysqli_fetch_array($result3))
        {
            $sql4="SELECT AuthorName FROM Authors WHERE
                AuthorID='$row3[0]';";
            $res4=mysqli_query($conn,$sql4);
            $row4=mysqli_fetch_row($res4);
            echo '<a href =
                "author.php?AuthorID='.$row3[0].'">'.$row4[0].'</a>,';
        }
        echo '</p></div></div>';
        $num++;
    }
    echo '</div>';
?>
```

4.5.3 Page Turning System

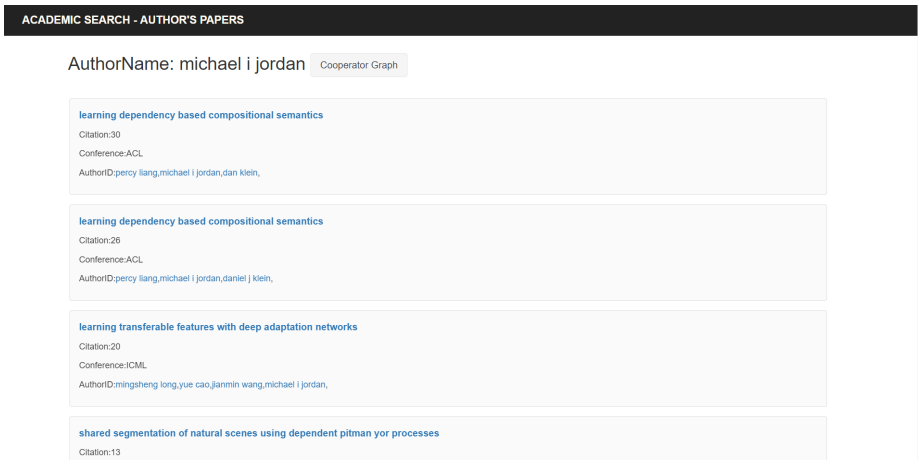
As here I do not arrange data in the form of the table, so I have to modify my page turning button. To differentiate different information in each item, I create a loop and label them in the form of 'XY'. X represents the position of information in corresponding item, and Y represents the postion of the item. Then it becomes quite easy to deal with it. The code is as below.

```
<script>
    function NextPage(AuthorID)
    {
        page = page + 1;
        $.ajax
        ({
            url:"author_hint.php",
            type:'post',
            dataType:"json",
            data:{'AuthorID':AuthorID,'Page':page},
            success:function (data)
            {
             for(var i = 0;i < data.length; i++)
                {
                    var row = data[i];
                    document.getElementById('1'+String(i+1)).innerHTML='<a
                        href="http://localhost/uikit/paper_front_ver2.php?PaperTitle="'+Strin
                    document.getElementById('2'+String(i+1)).innerHTML='Citation:'+String(row
                    document.getElementById('3'+String(i+1)).innerHTML='Conference:'+String(r
                    var aid =document.getElementById('4'+String(i+1));
                    aid.innerHTML = "AuthorName:";
                    for(var j = 0;j < row[5].length;j++)
                    {
                        aid.innerHTML = aid.innerHTML + "<a href =
                            'author.php?AuthorID="+String(row[5][j][0])+"'>"+String(row[6][j])
                    }


                }
            }
        })
    }
</script>

<?php
```

```
35      echo "<div style='margin:auto 7% auto'><h2>AuthorName: {$AuthorName}
            <button class=\"uk-button uk-button-large\"
            data-uk-offcanvas=\"{target:'#graph'}\">Cooperator
            Graph</button><h2></div>";
36      echo ' <div class="uk-grid" data-uk-grid-margin
            data-uk-scrollspy="{cls:\'uk-animation-fade uk-invisible\',
            target:\'> div > .uk-panel\', delay:300, repeat: true}">';
37      $num=1;
38      while($row1 = mysqli_fetch_array($result1))
39      {
40          echo '<div class="uk-width-medium-5-6" style="font-size:
                12px;margin: 1% 7% auto"><div class="uk-panel uk-panel-box
                uk-invisible"><p id="1'.$num.'" style="font-size:
                16px;font-weight: bolder"><a
                href="http://localhost/uikit/paper_front_ver2.php?PaperTitle='.$row1[0].'">'.$ro
                id="2'.$num.'">Citation:'.$row1[2].'</p></dd>';
41          $sql2 = "SELECT ConferenceName
42                FROM conferences
43                WHERE ConferenceID = '$row1[3]'
44                ";
45          $result2 = mysqli_query( $conn, $sql2 );
46
47          if(! $result2 )
48          {
49              die('Could not get data: ' . mysqli_error($conn));
50          }
51          $row2 = mysqli_fetch_array($result2);
52
53
54         echo '<dd><p id="3'.$num.'">Conference:'.$row2[0].'</p></dd><dd>';
55
56
57          $sql3 = "SELECT AuthorID
58                FROM paper_author_affiliation
59                WHERE PaperID = '$row1[1]'
60                ORDER BY AuthorSequence ASC";
61          $result3 = mysqli_query( $conn, $sql3 );
62
63
64          if(!$result3)
65          {
66              die('Could not get data: ' . mysqli_error($conn));
67          }
```

```
68          echo "<p id='4".$num."'>AuthorID:";
69          while($row3 = mysqli_fetch_array($result3))
70          {
71              $sql4="SELECT AuthorName FROM Authors WHERE
                    AuthorID='$row3[0]';";
72              $res4=mysqli_query($conn,$sql4);
73              $row4=mysqli_fetch_row($res4);
74              echo '<a href =
                    "author.php?AuthorID='.$row3[0].'">'.$row4[0].'</a>,';
75          }
76          echo '</p></div></div>';
77          $num++;
78      }
79      echo '</div>';
80  ?>
```

4.5.4 Results Show



5 SUMMARY

Actually, at the beginning of this final project, all of the tasks seem to be so formidable for us to have a try. Nearly all of the knowledge is unfamiliar and each line of code is elusive to us. But in the end, we actually did it. Thanks for every teacher who helps us and thanks for this class to give us such an unforgettable experience through the whole semester!