

# Experiment report of EE101 (Final)

Group 12

F1703014 517030910361 ZHOU LE

F1703013 517030910309 TANG JINGYAO

F1703014 517030910338 WU SHUSHU

F1703013 517030910308 MING YUXIN

June 16, 2018

## Contents

|          |                                           |           |
|----------|-------------------------------------------|-----------|
| <b>1</b> | <b>Experiment purpose and requirement</b> | <b>3</b>  |
| 1.1      | Purpose . . . . .                         | 3         |
| 1.2      | Basic Requirement . . . . .               | 3         |
| 1.3      | Advance Requirement . . . . .             | 3         |
| <b>2</b> | <b>Website Structure</b>                  | <b>4</b>  |
| 2.1      | Relationship among pages . . . . .        | 4         |
| 2.2      | Brief introduction of each page . . . . . | 4         |
| 2.2.1    | home.php . . . . .                        | 4         |
| 2.2.2    | result.php . . . . .                      | 5         |
| 2.2.3    | author.php . . . . .                      | 5         |
| 2.2.4    | title.php . . . . .                       | 6         |
| 2.2.5    | paper.php . . . . .                       | 6         |
| 2.2.6    | conferenceresult.php . . . . .            | 7         |
| 2.2.7    | conference.php . . . . .                  | 7         |
| 2.2.8    | tree.php . . . . .                        | 8         |
| 2.2.9    | hint.php . . . . .                        | 8         |
| 2.2.10   | error.php . . . . .                       | 9         |
| 2.2.11   | produce.html . . . . .                    | 9         |
| <b>3</b> | <b>Website Features</b>                   | <b>10</b> |
| 3.1      | CI framework . . . . .                    | 10        |
| 3.1.1    | Introduction . . . . .                    | 10        |
| 3.1.2    | Advantages . . . . .                      | 11        |
| 3.2      | Auto-completion . . . . .                 | 11        |
| 3.2.1    | Search model . . . . .                    | 11        |
| 3.2.2    | Search . . . . .                          | 12        |
| 3.2.3    | Home . . . . .                            | 14        |
| 3.2.4    | Result . . . . .                          | 15        |
| 3.3      | Paper Recommend . . . . .                 | 15        |
| 3.3.1    | Analysis . . . . .                        | 15        |
| 3.3.2    | Codes . . . . .                           | 15        |
| 3.3.3    | Result . . . . .                          | 17        |
| 3.4      | Force-directed Graph . . . . .            | 17        |
| 3.4.1    | Introduction . . . . .                    | 17        |
| 3.4.2    | Implementation . . . . .                  | 17        |

|          |                                                    |           |
|----------|----------------------------------------------------|-----------|
| 3.5      | Tree Graph . . . . .                               | 17        |
| 3.5.1    | Introduction . . . . .                             | 17        |
| 3.5.2    | Implementation . . . . .                           | 18        |
| 3.6      | Support functions . . . . .                        | 22        |
| 3.6.1    | Introduction . . . . .                             | 22        |
| 3.6.2    | When no searching result . . . . .                 | 22        |
| 3.6.3    | When turn to nonexistent page . . . . .            | 22        |
| <b>4</b> | <b>Database Designed</b>                           | <b>23</b> |
| 4.1      | Citation . . . . .                                 | 23        |
| 4.1.1    | Analysis . . . . .                                 | 23        |
| 4.1.2    | Codes . . . . .                                    | 23        |
| 4.1.3    | Result . . . . .                                   | 24        |
| 4.2      | SPEED-CACHE . . . . .                              | 25        |
| 4.2.1    | To speed up . . . . .                              | 25        |
| 4.2.2    | Codes . . . . .                                    | 25        |
| <b>5</b> | <b>Beautification</b>                              | <b>25</b> |
| 5.1      | Design for home . . . . .                          | 25        |
| 5.2      | Design For Producer . . . . .                      | 30        |
| 5.3      | Design for Result . . . . .                        | 32        |
| 5.4      | Author . . . . .                                   | 38        |
| 5.5      | 404 page . . . . .                                 | 40        |
| <b>6</b> | <b>Optimization</b>                                | <b>41</b> |
| 6.1      | Introduction . . . . .                             | 41        |
| 6.2      | An example of running speed Optimization . . . . . | 41        |
| 6.3      | An example of code Optimization . . . . .          | 42        |
| <b>7</b> | <b>Conclusion</b>                                  | <b>45</b> |

# **1 Experiment purpose and requirement**

## **1.1 Purpose**

- Integrate the previous website system.
- Improve the function.
- Optimize the interface.

## **1.2 Basic Requirement**

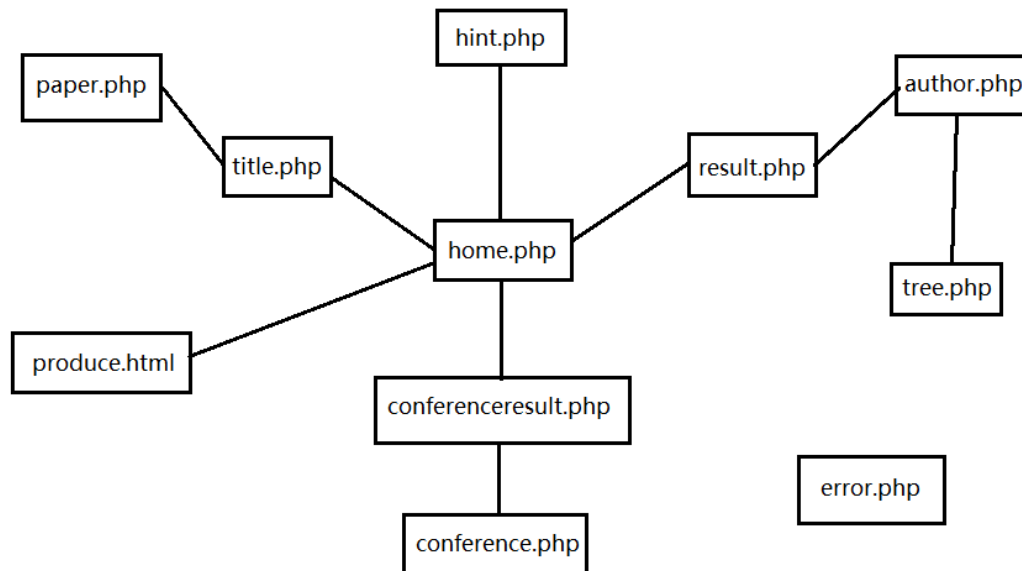
- The search results page adds searches for paper title and conference.
- Add paper and conference pages to website.
- When displaying information on each page, if more information items are displayed , then 10 items are displayed on each page.
- Add page-turning function.

## **1.3 Advance Requirement**

- Realize paper recommending function
- The search results are no longer based on simple mysql queries, but can be implemented with solr or elasticsearch.
- The tree structure of teacher relationship can be displayed, and the tree can be opened layer by layer.
- Website performance acceleration, database optimization, SQL statement optimization, etc.

## 2 Website Structure

### 2.1 Relationship among pages



Attention: Exclude supporting pages of page-turning functions

Figure 1: 2.1

### 2.2 Brief introduction of each page

#### 2.2.1 home.php

Functions:

- Main page of the website, each page can turn back to it.
- Provide an input box and buttons to get and send user's intentions of search.
- Provide auto completion function when inputting.
- Provide a connection to show all stuff.

Displays:

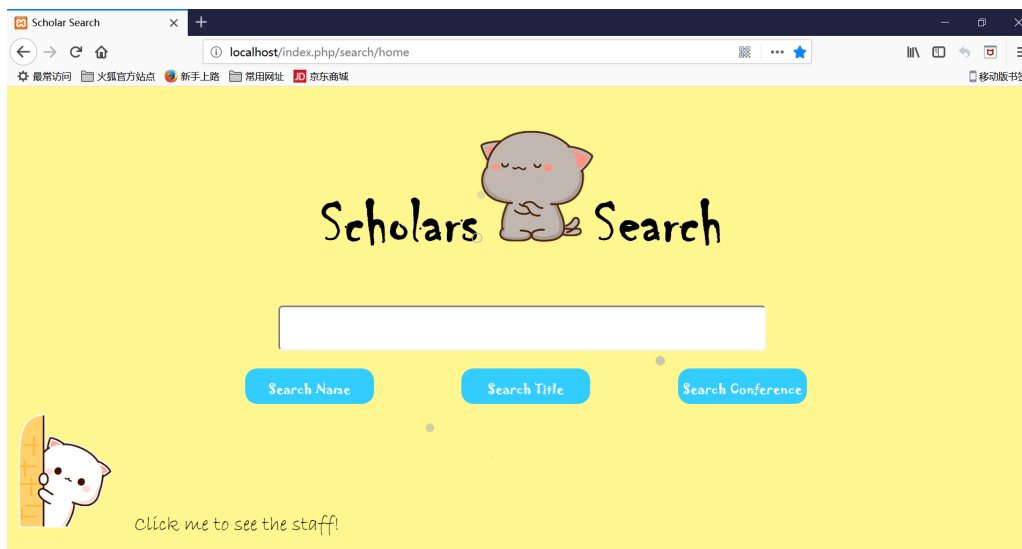


Figure 2: 2.2.1

### 2.2.2 result.php

Function:

- Show result of author-searching.
- Page turning with AJAX and SQL cache.

Displays:

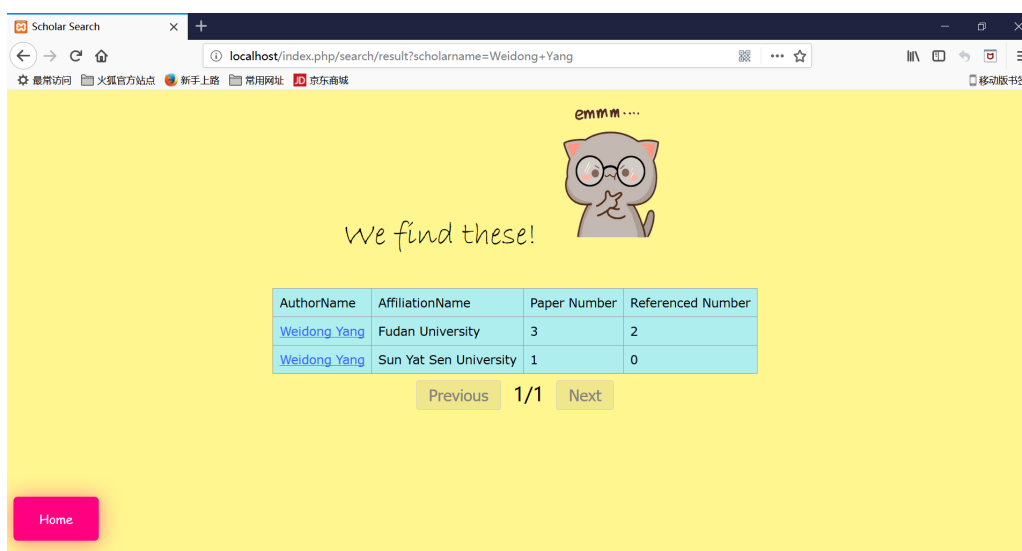


Figure 3: 2.2.2

### 2.2.3 author.php

Function:

- Show detail information of selected author, including his basic information and papers of him.
- Show force-directed graph about the selected author between his advisors and between his students.
- Connection to tree graph page of selected author.

- Page turning with AJAX and SQL cache.

Displays:

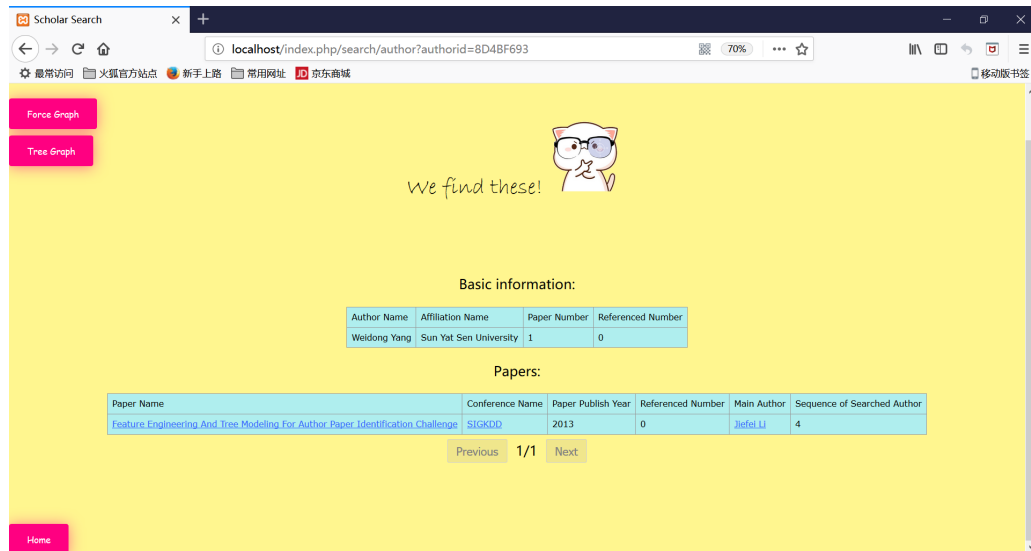


Figure 4: 2.2.3

## 2.2.4 title.php

Function:

- Show result of paper-searching.
- Page turning with AJAX and SQL cache.

Displays:

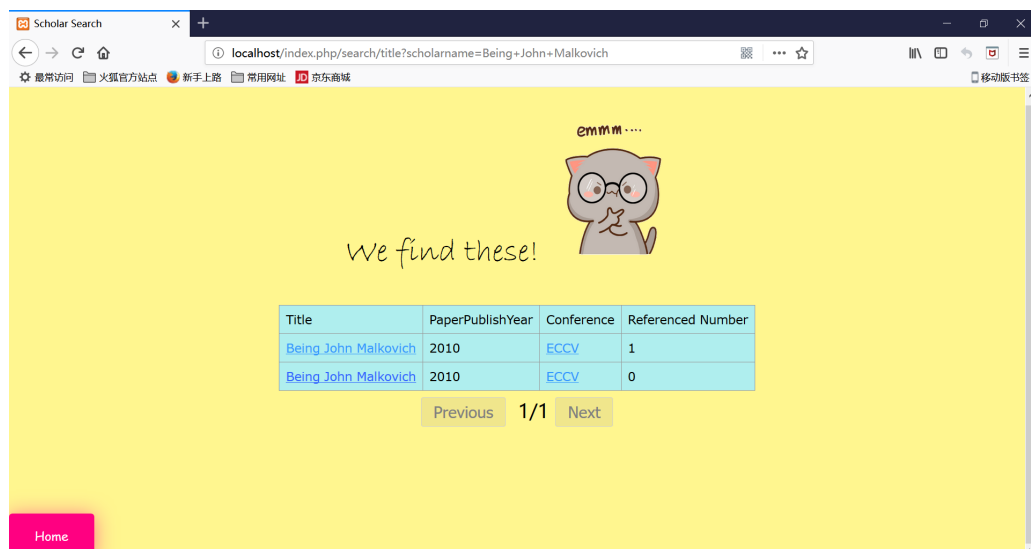


Figure 5: 2.2.4

## 2.2.5 paper.php

Function:

- Show detail information of selected paper, including its basic information and authors of it.

- Show recommend papers of it if it has.
- Page turning with AJAX and SQL cache.

Displays:

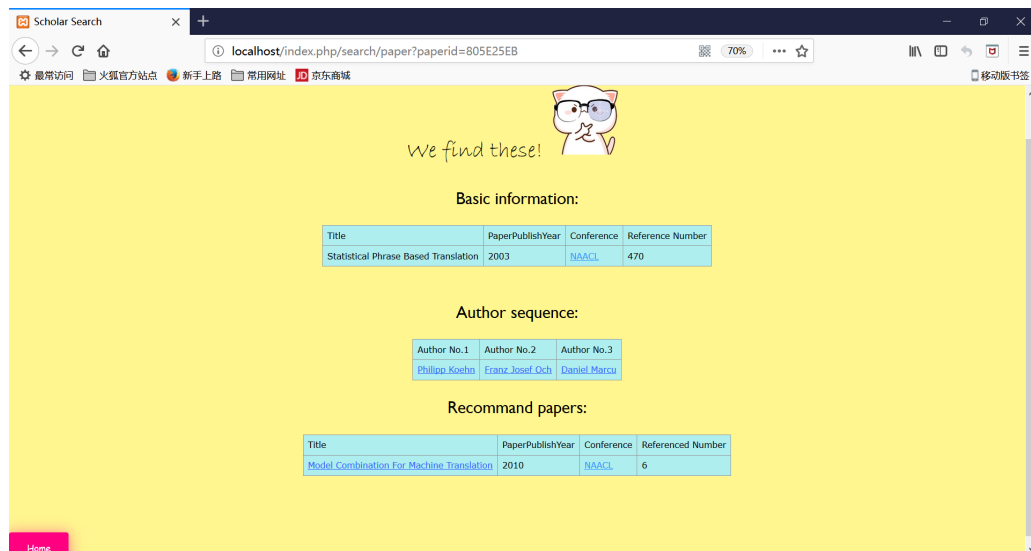


Figure 6: 2.2.5

### 2.2.6 conferenceresult.php

Function:

- Show result of conference-searching.

Displays:

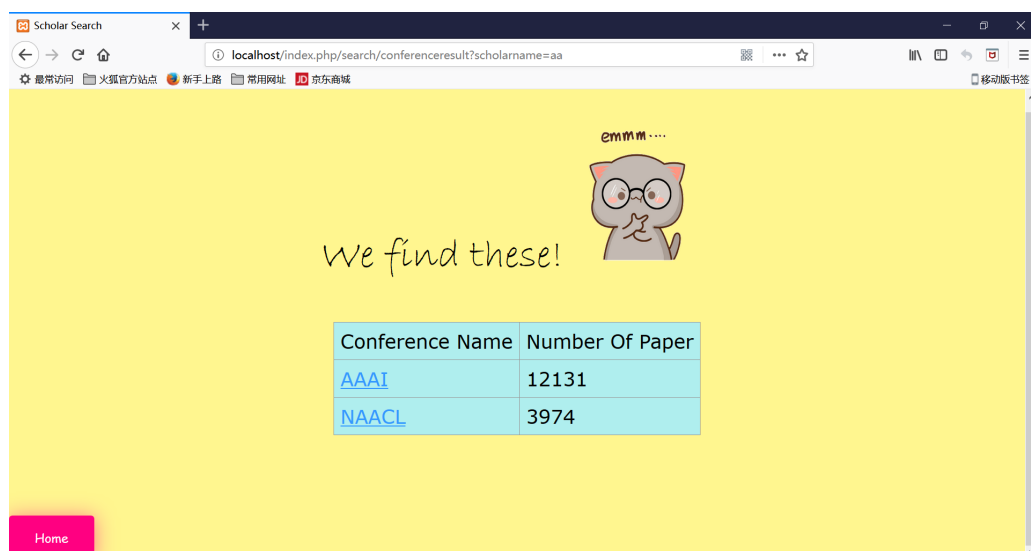


Figure 7: 2.2.6

### 2.2.7 conference.php

Function:

- Show detail information of selected paper, including its paper number and all papers of it.

- Page turning with AJAX and SQL cache.

Displays:

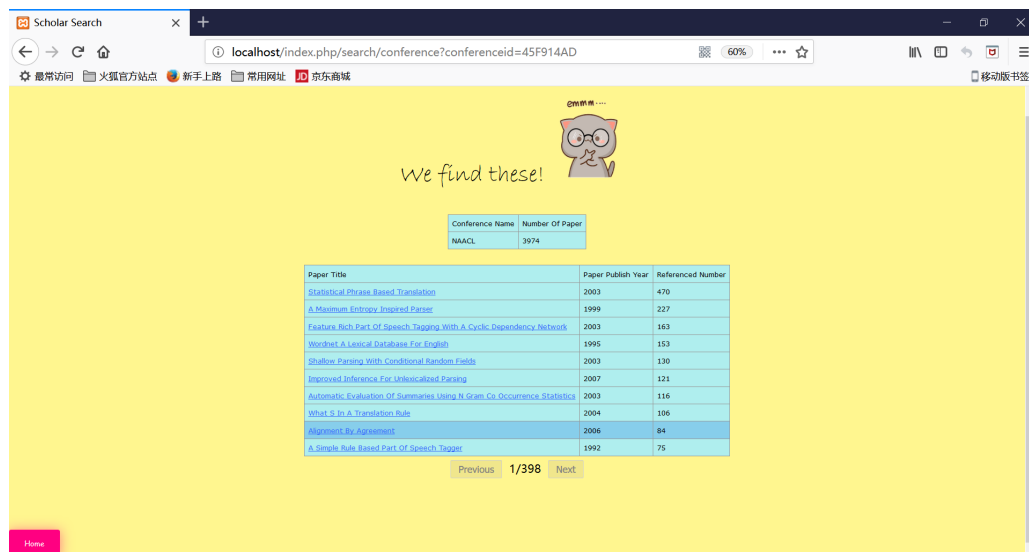


Figure 8: 2.2.7

## 2.2.8 tree.php

Function:

- Show tree graph about advisors and students of selected author.
- Provide dynamic update function when click the names.

Displays:

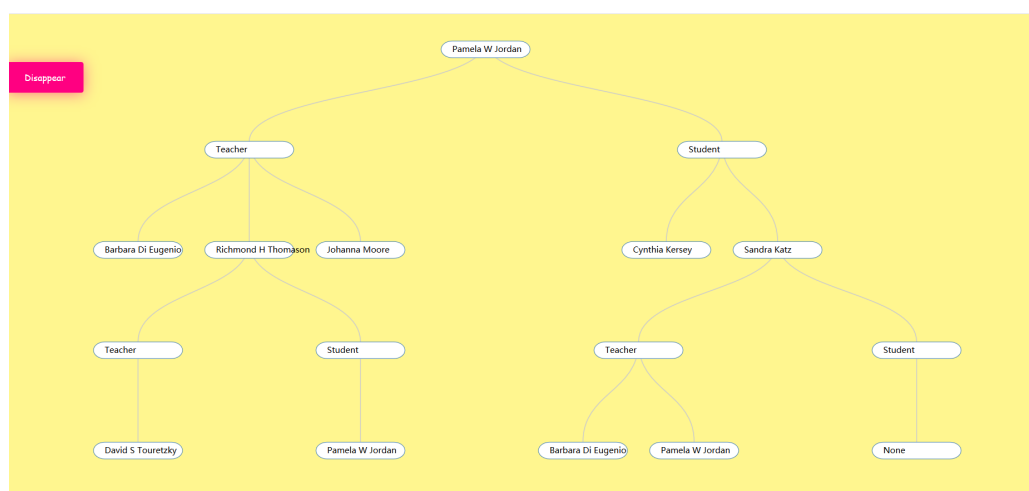


Figure 9: 2.2.8

## 2.2.9 hint.php

Function:

- Give author, paper and conference recommend when inputting several letters by connecting to the database.

Displays: None

### 2.2.10 error.php

Function:

- Give feed back when no result of current search.

Displays:

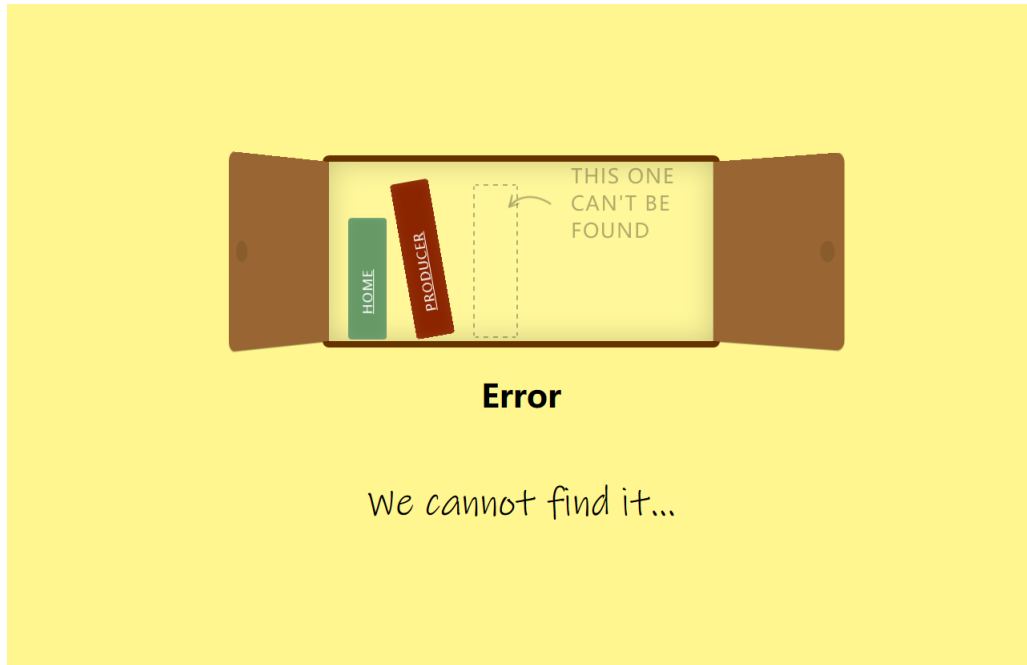


Figure 10: 2.2.9

### 2.2.11 produce.html

Function:

- Show the group members of the website.

Displays:

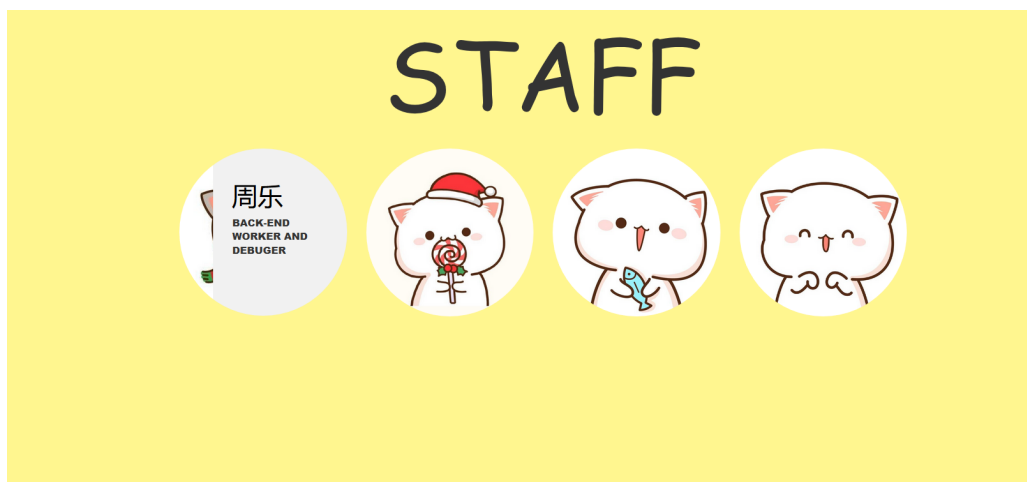


Figure 11: 2.2.10

## 3 Website Features

### 3.1 CI framework

#### 3.1.1 Introduction

CI framework is a kind of MVC separated website framework. A classic type of CI framework is like follow:

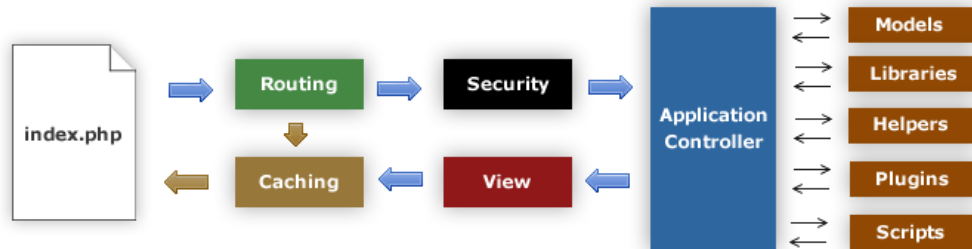


Figure 12: 3.1.1

For easier to build, our website is based on CI framework, and we put prototypes of all the pages into controller, put all the SQL statement into model, and put front-end with its beautification code into views of each page.

For example, suppose now we need to get the keyword from home.php to result.php and get the result to show.

First, we can write a function into controller in CI controller Class "Search" like follow:

```
1 public function result()
2 {
3     $s = $_GET["scholarname"];
4     if(strlen($s)<2)
5     {
6         $this->load->view('search/error');
7         return;
8     }
9     $query = $this->search_model->get_all_authorresult($s);
10    if(!count($query))
11    {
12        $this->load->view('search/error');
13        return;
14    }
15    $alldatanum=count($query);
16    if($alldatanum>10)
17    $datanum=10;
18    else
19    $datanum=$alldatanum;
20    $data=array();
21    $query==array_slice($query,0,$datanum);
22    foreach ($query as $row)
23    {
24        $query2 = $this->search_model->get_affiliation($row['AuthorID']);
25        if(count($query2)==0)
26        $query2[]=array('AffiliationName'=>"");
27        foreach($query2 as $row2)
28        {
29            $data[]=array($row,$row2);
30        }
31    }
32    $trans =array(
33        'data' => $data,
34        'datanum' => $datanum,
35        'scholarname' => $s,
36        'alldatanum' => $alldatanum
37    );
38    $this->load->view('search/result',$trans);
```

This code means, get the word by GET from other page, and check if it is less than 2 word. If so, load "error" page, if not, connect to database to get the result. If it has result, continue to process, otherwise load "error" page. After all the process, it will send all the data by an associative array to corresponding "view" php, and create varieties there.

In view of result, it will be the same as tradition php web site, but it only needs to care about how to present the data.

### 3.1.2 Advantages

- As you can see, CI framework has good encapsulation, we do not need to care about how other part realize when we focus on one of the part.
- Because every part of it is separated, we can add anything with any sequence in controller and model, each group member can focus on different thing respectively, and improve the efficiency.
- Easier to locate and fix bugs.
- Easier to manage all the pages.

## 3.2 Auto-completion

### 3.2.1 Search model

Get the required amount of datas to complete automatically

```
public function get_five_author($s)
{
    $query = $this->db->query("select authors.AuthorID,authors.AuthorName
from authors
inner join paper_author_affiliation
on paper_author_affiliation.AuthorID=authors.AuthorID
where authors.AuthorName like \"%{$s}%\"
group by(authors.AuthorID)
order by count(paper_author_affiliation.PaperID)DESC
limit 0,5");
    return $query->result_array();
}

public function get_three_conference($s)
{
    $query = $this->db->query("select conferences.ConferenceID,conferences.ConferenceName
from conferences
inner join papers
on papers.ConferenceID=conferences.ConferenceID
where conferences.ConferenceName like \"%{$s}%\"
group by(conferences.ConferenceID)
order by count(papers.PaperID)DESC
limit 0,3");
    return $query->result_array();
}

public function get_five_title($s)
{
    $query = $this->db->query("select papers.PaperID,papers.Title
from papers inner join paper_reference on papers.PaperID=paper_reference.PaperID
where papers.Title like \"%{$s}%\"
group by(paper_reference.PaperID)
order by count(paper_reference.ReferenceID)DESC
limit 0,5");
    return $query->result_array();
}
```

```
}
```

Analysis:

Write three functions to get five authors, five papers' titles and three conferences' names according to the letters entered by users using fuzzy search.

Authors are ordered by the amount of papers that they have written. Papers are ordered by the amount of references. Conferences are ordered by the amount of papers that are related to the conference.

### Get more information about each row

```
public function get_all_papers($authorid)
{
    $query = $this->db->query("select papers.PaperID,papers.Title
    from papers
    inner join paper_author_affiliation
    on paper_author_affiliation.PaperID=papers.PaperID
    where paper_author_affiliation.AuthorID = \"{$authorid}\"
    group by(papers.PaperID)
    order by count(papers.ConferenceID)DESC");
    return $query->result_array();
}

public function get_all_conferences($conferenceid)
{
    $query = $this->db->query("select PaperID from papers
    where ConferenceID=\"{$conferenceid}\" ");
    return $query->result_array();
}

public function get_all_references($paperid)
{
    $query = $this->db->query("select ReferenceID from paper_reference
    where PaperID=\"{$paperid}\" ");
    return $query->result_array();
}
```

Analysis:

Write three functions to get more information about each automatic complement.

They are used to find the amount of papers that the author have published, the amount of all references that are about the paper and the amount of all papers that are related to the conference.

### 3.2.2 Search

#### Authors' complement

```
$q = ($_GET["term"]);
$aquery = $this->search_model->get_five_author($q);
foreach ($aquery as $row)
{
    $num1 = $this->search_model->get_all_papers($row['AuthorID']);
    $res1 = count($num1);
    $result[] = array(
        'id' =>ucwords($row['AuthorName']),
        'label' => "Author:".ucwords($row['AuthorName'])."(".$res1." papers)"
    );
}
```

Analysis:

Use the function get five author written in Searchmodel.php, create a variable named aquery to store the result.

For every element in aquery named row, find the related amount of papers by using the function named get all papers written in Searchmodel.php and store it in variable named num1.  
Create a variable named result, which has two elements id and label storing the name of the author, the name and the related numbers respectively.

### Conferences' complement

```
$cquery = $this->search_model->get_three_conference($q);  
foreach ($cquery as $row)  
{  
    $num3 = $this->search_model->get_all_conferences($row['ConferenceID']);  
    $res3 = count($num3);  
    $result[] = array(  
        'id' =>ucwords($row['ConferenceName']),  
        'label' => "Conference:".ucwords($row['ConferenceName'])."(".$res3." papers)"  
    );  
}
```

Analysis:

The same with the authors' complement.

### Papers' complement

```
$cquery = $this->search_model->get_three_conference($q);  
foreach ($bquery as $row)  
{  
    $num2 = $this->search_model->get_all_references($row['PaperID']);  
    $res2 = count($num2);  
    $Length=strlen($row['Title']);  
    if($Length>30)  
    {  
        $num=substr($row['Title'],0,30)."...";  
        $result[] = array(  
            'id' =>ucwords($row['Title']),  
            'label' => "Title:".ucwords($num)."(".$res2." references)"  
        );  
    }  
    else  
    {  
        $result[] = array(  
            'id' =>ucwords($row['Title']),  
            'label' => "Title:".ucwords($row['Title'])."(".$res2." references)"  
        );  
    }  
}
```

Analysis:

The title of a paper may be too long to show on the screen. So it is a little bit different from above.

Use function strlen() to get the length of title and create a variable named Length to store the length of title. If it's larger than 30, use function substr() to store the first thirty letters of title in variable named num. The variable named result has two elements id and label. Element id is used to store the title completely. Element label is used to store the substring of title and the amount of references.

### Return the result:

```
echo json_encode($result);
```

Analysis:

Return the result of searching in the json form.

### 3.2.3 Home

Basic settings of Auto complement:

```
$( "#scholarname" ).autocomplete({  
    source: "hint",  
    minLength: 2,
```

Analysis:

Name the asked one as "scholarname".

Set the data's source is function named "hint" written in Search.php.

Set the smallest length of the input to do auto complement.

Select

```
select: function( event, ui ) {  
    $( "#scholarname" ).val( ui.item.id );  
    return false;  
},
```

Analysis:

The function is used to set the content showed in input box when the mouse is put on the corresponding row. The parameter is result's element named id.

Focus

```
focus: function( event, ui ) {  
    $( "#scholarname" ).val( ui.item.id );  
    return false;  
},
```

Analysis:

Without this function, when the user click one row showed below, the corresponding content will be showed in input box with useless information, when the user click search, the system will treat the whole content as author's name or conference's name by mistake, so it can't find any information at last.

The function is used to modify the content filled in input box when the user click one of the rows below. Its parameter is result's element named id, which stores the really useful information.

### 3.2.4 Result

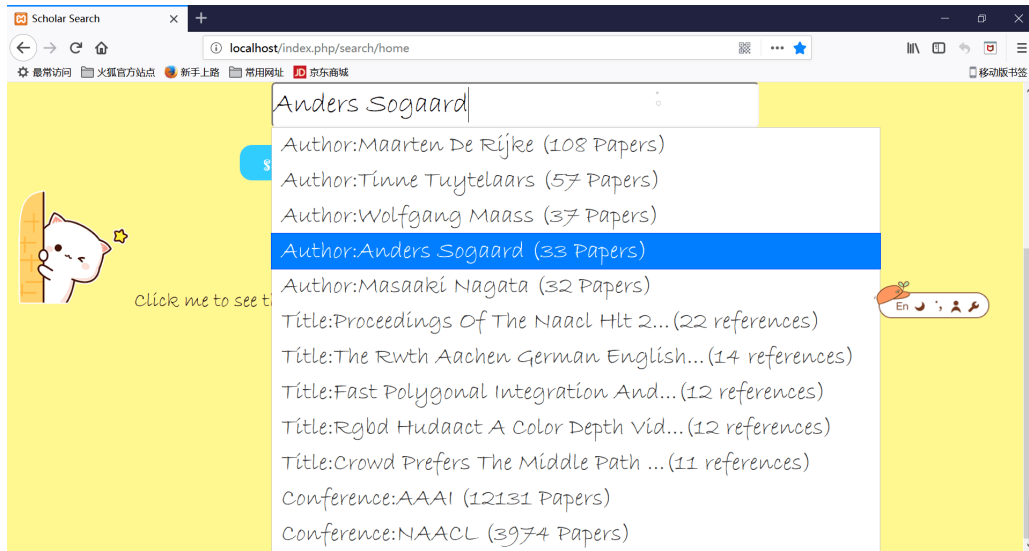


Figure 13: 2

## 3.3 Paper Recommend

For each paper, we need to recommend another related paper to the reader. So we need create another table, where the first column is PaperID A, and the second is A's related PaperID.

### 3.3.1 Analysis

- How to define whether two papers are related?
  - The two papers need to have at least one common author.
  - For paper A, if paper B has cited paper A, then paper B is related to paper A
  - If paper A and paper B have ever cited a common paper, then A and B are related.
- How to realize it? Write three functions, MayConWith, Cite, conn.
  - MayConWith: This function gets one parameter id, and returns all the papers the id's authors have written.
  - Cite: This function gets one parameter id, and returns all the papers this paper has cited.
  - conn: This function gets two parameters id a and b, and returns whether b is related to a.
- How to create the table? First use a basic create table sql statement to create the table connection. Second call the three functions and use the insert statement to add the information into the table.

### 3.3.2 Codes

- For the three functions: (connectionOrNot.py)

```
1  #the first function:
2  def MayConWith(paperid):
3      l=[]
4      sql1="""SELECT AuthorID from paper_author_affiliation
5              WHERE PaperID="{}"
6              """.format(paperid)
7
8      try:
9
10         cursor.execute(sql1)
11
```

```

12     results = cursor.fetchall()
13     for row in results:
14         authorid=row[0]
15         sql2 = """SELECT PaperID from paper_author_affiliation
16                 WHERE AuthorID="{0}" and PaperID!="{0}"
17                 """.format(authorid,paperid)
18         try:
19             cursor.execute(sql2)
20             results2 = cursor.fetchall()
21             for row2 in results2:
22                 if row2[0] not in l:
23                     l.append(row2[0])
24         except:
25             print("Error")
26     except:
27         print("Error: unable to fetch data")
28     return l
29 #the second function:
30 def Cite(idA):
31     l=[]
32     sql3="""SELECT ReferenceID from paper_reference
33           WHERE PaperID ="{}"
34           """.format(idA)
35     cursor.execute(sql3)
36     results3 = cursor.fetchall()
37     for row in results3 :
38         l.append(row[0])
39     return l
40 #the third function:
41 def conn(idA,idB):
42     lb=Cite(idB)
43     if idA in lb:
44         return True
45     la=Cite(idA)
46     for i in lb:
47         if i in la:
48             return True
49     return False

```

---

## 2. To create the table: (create\_Table.py)

```

1  from connectionOrNot import MayConWith,conn
2  import pymysql
3  db = pymysql.connect(host='localhost',
4                       user='root',
5                       password='',
6                       db='main_db',
7                       charset='utf8',
8                       port=3306,
9                       cursorclass=pymysql.cursors.Cursor)
10 cursor=db.cursor()
11 sql1=""" SELECT PaperID from papers """
12 cursor.execute(sql1)
13 results=cursor.fetchall()
14 for row in results:
15     paperid=row[0]
16     l=MayConWith(paperid)
17     for i in l:
18         if conn(paperid,i):
19             sql2 = """INSERT INTO conection(PaperID,ConPaperID)
20                     VALUES("{0}", "{0}")""".format(paperid,i)
21             try:
22                 cursor.execute(sql2)
23                 db.commit()
24             except:
25                 print("error!!!!!!")
26                 db.rollback()
27
28 db.close()

```

---

### 3.3.3 Result

- The connection table:

| PaperID  | ConPaperID |
|----------|------------|
| 0017A546 | 77FB4379   |
| 0017A546 | 78BD7FC2   |
| 0017A546 | 80FA2D50   |
| 0017A546 | 6DFDD723   |
| 001DEA68 | 0EEA8FF5   |
| 001DEA68 | 7A2CB77D   |

- The page:

| Recommend papers:                                                                                     |                  |                     |                   |
|-------------------------------------------------------------------------------------------------------|------------------|---------------------|-------------------|
| Title                                                                                                 | PaperPublishYear | Conference          | Referenced Number |
| <a href="#">Profile Based Cross Document Coreference Using Kernelized Fuzzy Relational Clustering</a> | 2009             | <a href="#">ACL</a> | 1                 |

## 3.4 Force-directed Graph

### 3.4.1 Introduction

The force-directed graph can show the relationship between authors like experiment4, here we add some new features to it. First, it can tell you what do these colors of points mean. Second, it uses AuthorID to differentiate each point instead of AuthorName, and AuthorName can be used as the label of each point, which prevents bugs when authors with the same name show at the same time.

### 3.4.2 Implementation

Suppose we have a normal force-directed graph as experiment4's (here we do not show it in detail), in the view of author page, we can modify the following js code to give fixed color of author with different group (main author, advisor, student, irrelative):

```
1 var node = svg_force.append("g")
2   .attr("class", "nodes")
3   .selectAll("circle")
4   .data(nodes)
5   .enter().append("circle")
6     .attr("r", 5)
7     .attr("fill", function(d) {
8       if(d.group==1) return "#333399";
9       if(d.group==2) return "#ff3366";
10      if(d.group==3) return "#66ccff";
11      if(d.group==4) return "#ffcc99";
12    })
13     .call(d3.drag()
14       .on("start", dragstarted)
15       .on("drag", dragged)
16       .on("end", dragended));
```

To the second features, we can also add this code in "view" to show each point with their name instead of their id:

```
1 node.append("title")
2   .text(function(d) { return d.name; });
```

## 3.5 Tree Graph

### 3.5.1 Introduction

The tree graph can show all teachers and students of every author by dynamic updating, and it can be fold, unfold and dragged with animation effects.

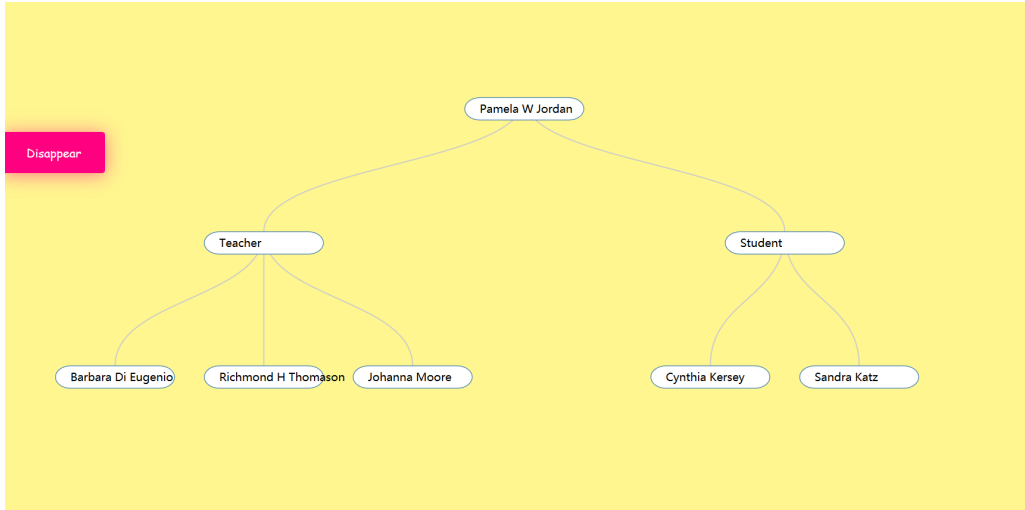


Figure 14: 3.5.1-1

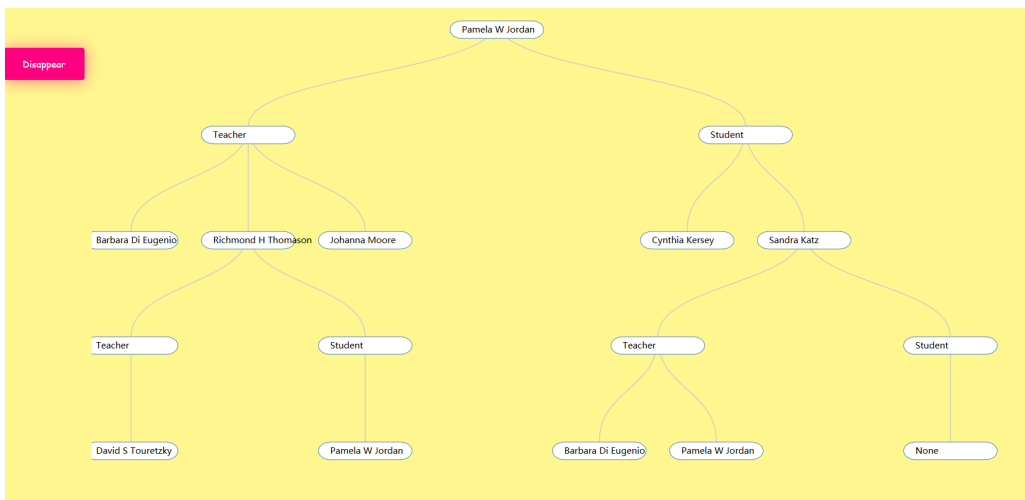


Figure 15: 3.5.1-2

### 3.5.2 Implementation

(As it uses different version of d3 library, we need to build it on another page. And we can use a button to jump between it and author.php invisibly.) First, we need to get initial data when first open the graph. Following code creates a data structure of tree. Each node have its id (be used to identify), its name (be used to show in screen) and its children nodes (if has). For the main searching author, it will be the root node and have two children nodes—teacher and student. And these two children nodes will store what exactly the students and teachers are.

```

1 public function tree()
2 {
3     $id = $_GET["authorid"];
4     $mainauthor = $this->search_model->get_id_author($id)[0]['AuthorName'];
5     $root=array(
6         'name'=> ucwords($mainauthor),
7         'label' => $id,
8         'children'=> array(
9             array('name'=>'Teacher', 'children'=>array()),
10            array('name'=>'Student', 'children'=>array())
11        )
12    );
  
```

```

13     $query=$this->search_model->get_advisors_newtree($id);
14     $query2=$this->search_model->get_students_newtree($id);
15     if(count($query)==0) $root['children'][0]['children'][]=array('name'=>'None');
16     else
17     foreach($query as $row)
18     {
19         $query3=$this->search_model->get_id_author($row['advisorID']);
20         $root['children'][0]['children'][]=array('name'=>ucwords($query3[0]['AuthorName']), 'label'=>$row['advisorID']);
21     }
22     if(count($query2)==0) $root['children'][1]['children'][]=array('name'=>'None');
23     else
24     foreach($query2 as $row)
25     {
26         $query3=$this->search_model->get_id_author($row['studentID']);
27         $root['children'][1]['children'][]=array('name'=>ucwords($query3[0]['AuthorName']), 'label'=>$row['studentID']);
28     }
29     $trans=array(
30         'authorid' => $id,
31         'root' => $root
32     );
33     $this->load->view('search/tree',$trans);
34 }

```

Second, we should show the nodes as "tree" by d3 in front-end. Like exp4, we should create a svg element and modify some characteristics of nodes to show nodes' name and let it have appropriate size and animation effects.

```

1  <script>
2
3  var root = <?php echo json_encode($root);?>;
4  var margin = {top: 40, right: 120, bottom: 20, left: 700},
5              width = 1600,
6              height = 1600;
7
8  var i = 0,
9  duration = 750,
10 root;
11
12 var tree = d3.layout.tree().nodeSize([200, 0]);
13 var diagonal = d3.svg.diagonal()
14     .projection(function(d) { return [d.x, d.y]; });
15
16
17 var svg = d3.select("svg")
18     .append("g")
19     .call(zm = d3.behavior.zoom().scaleExtent([1,3]).on("zoom", redraw)).append("g")
20     .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
21
22 function redraw() {
23     svg.attr("transform", "translate(" + d3.event.translate + ")" + " scale(" + d3.event.scale + ")");
24 }
25
26
27 zm.translate([512, 50]);
28
29 root.x0 = 0;
30 root.y0 = height / 2;
31
32 function collapse(d) {
33     if (d.children) {
34         d._children = d.children;
35         d._children.forEach(collapse);
36         d.children = null;
37     }
38 }
39
40 root.children.forEach(collapse);
41 update(root);
42
43
44 d3.select(self.frameElement).style("height", "800px");
45
46 function update(source) {
47

```

```

48 // Compute the new tree layout.
49 var nodes = tree.nodes(root).reverse(),
50     links = tree.links(nodes);
51
52 // Normalize for fixed-depth.
53 nodes.forEach(function(d) { d.y = d.depth * 180; });
54
55 // Update the nodes
56 var node = svg.selectAll("g.node")
57     .data(nodes, function(d) { return d.id || (d.id = ++i); });
58
59 // Enter any new nodes at the parent's previous position.
60 var nodeEnter = node.enter().append("g")
61     .attr("class", "node")
62     .attr("transform", function(d) { return "translate(" + source.x0 + "," + source.y0 + ")"; })
63     .on("click", click);
64
65 nodeEnter.append("rect")
66     .attr("x", "-80")
67     .attr("rx", "20")
68     .attr("ry", "20")
69     .attr("width", 1e-6)
70     .attr("height", 1e-6)
71     .style("fill", function(d) { return d._children ? "#ccccff" : "#fff"; });
72
73 nodeEnter.append("text")
74     .attr("x", function(d) { return -60; })
75     .attr("y", "20")
76     .attr("text-anchor", function(d) { return "start"; })
77     .text(function(d) { return d.name; })
78     .style("fill-opacity", 1e-6);
79
80 // Transition nodes to their new position.
81 var nodeUpdate = node.transition()
82     .duration(duration)
83     .attr("transform", function(d) { return "translate(" + d.x + "," + d.y + ")"; });
84
85 nodeUpdate.select("rect")
86     .attr("width", 160)
87     .attr("height", 30)
88     .style("fill", function(d) { return d._children ? "#ccccff" : "#fff"; });
89
90 nodeUpdate.select("text")
91     .style("fill-opacity", 1);
92
93 // Transition exiting nodes to the parent's new position.
94 var nodeExit = node.exit().transition()
95     .duration(duration)
96     .attr("transform", function(d) { return "translate(" + source.x + "," + source.y + ")"; })
97     .remove();
98
99 nodeExit.select("rect")
100     .attr("width", 1e-6)
101     .attr("height", 1e-6);
102
103 nodeExit.select("text")
104     .style("fill-opacity", 1e-6);
105
106 // Update the links
107 var link = svg.selectAll("path.link")
108     .data(links, function(d) { return d.target.id; });
109
110 // Enter any new links at the parent's previous position.
111 link.enter().insert("path", "g")
112     .attr("class", "link")
113     .attr("d", function(d) {
114         var o = {x: source.x0, y: source.y0};
115         return diagonal({source: o, target: o});
116     });
117
118 // Transition links to their new position.
119 link.transition()
120     .duration(duration)

```

```

121     .attr("d", diagonal);
122
123     // Transition exiting nodes to the parent's new position.
124     link.exit().transition()
125         .duration(duration)
126         .attr("d", function(d) {
127             var o = {x: source.x, y: source.y};
128             return diagonal({source: o, target: o});
129         })
130         .remove();
131
132     // Stash the old positions for transition.
133     nodes.forEach(function(d) {
134         d.x0 = d.x;
135         d.y0 = d.y;
136     });
137 }
138 </script>

```

---

Third, to realize dynamic update, we should use ajax to update the origin tree if necessary.

We can add these code to front-end to create ajax event when click. These function can judge if it is necessary to ask for new data or to change the page, (for example, the "none" node can not be clicked and updata as it is meaningless).

```

1  <script>
2  function getNode(id){
3      var mynodes = null;
4      $.ajax({
5          url : "newtreeöauthorid="+id,
6          async : false, //
7          type : "GET",
8          dataType : "json",
9          success : function(data) {
10             mynodes = data;
11             console.log(mynodes);
12             //nodes = JSON.parse(nodes);
13         }
14     });
15     return mynodes;
16 }
17
18 // Toggle children on click.
19 function click(d) {
20     if (d.children) {
21         d._children = d.children;
22         d.children = null;
23     } else if (d._children){
24         d.children = d._children;
25         d._children = null;
26     } else {
27         if(d.name=="None") var mnodes = getNode(d.name);
28         else var mnodes=getNode(d.label);
29         if(mnodes==null) return;
30         d.children = mnodes.children;
31     }
32     update(d);
33 }
34 </script>

```

---

Then, we should build a support website in back-end to process "updating". It is almost the same as processing the origin data in step 1.(Here "pause" means the name of author who needs to show his teachers and students.)

```

1  public function newtree()
2  {
3      $id = $_GET["authorid"];
4      if($id=='None') return;
5      $query=$this->search_model->get_advisors_newtree($id);
6      $query2=$this->search_model->get_students_newtree($id);
7      $trans=array('name'=>'Pause', 'children'=>array(
8          array('name'=>'Teacher', 'children'=>array()),
9          array('name'=>'Student', 'children'=>array())
10     ), 'label'=>$id);

```

```

11     if(count($query)==0) $trans['children'][0]['children']=array('name'=>'None');
12     else
13     foreach($query as $row)
14     {
15         $query3=$this->search_model->get_id_author($row['advisorID']);
16         $trans['children'][0]['children']=array('name'=>ucwords($query3[0]['AuthorName']), 'label'=>$row['advisorID']);
17     }
18     if(count($query2)==0) $trans['children'][1]['children']=array('name'=>'None');
19     else
20     foreach($query2 as $row)
21     {
22         $query3=$this->search_model->get_id_author($row['studentID']);
23         $trans['children'][1]['children']=array('name'=>ucwords($query3[0]['AuthorName']), 'label'=>$row['studentID']);
24     }
25     echo json_encode($trans);
26 }

```

---

Finally, the tree graph can run well.

## 3.6 Support functions

### 3.6.1 Introduction

In some case, such as no searching result or unexpect operation of users', we want to give some special feed back instead of page with nothing or inactivity. Therefore, we add some code to our website to identify and beautify these situation.

### 3.6.2 When no searching result

In each search result page, we add these code in their controller.

```

1  $query = $this->search_model->get_all_authorresult($s);
2  if(!count($query))
3  {
4      $this->load->view('search/error');
5      return;
6  }

```

---

To have better performance, we can judge if the keywords are given longer than two letter, and return error feedback immediately when not in order to prevent exceeding maxium search time and long-time waiting.

```

1  $s = $_GET["scholarname"];
2  if(strlen($s)<2)
3  {
4      $this->load->view('search/error');
5      return;
6  }

```

---

### 3.6.3 When turn to nonexistent page

When you are at the first page to try to visit previous page or at the last page to try to visit next page, our front-end will identify this situation and show a special picture to you.

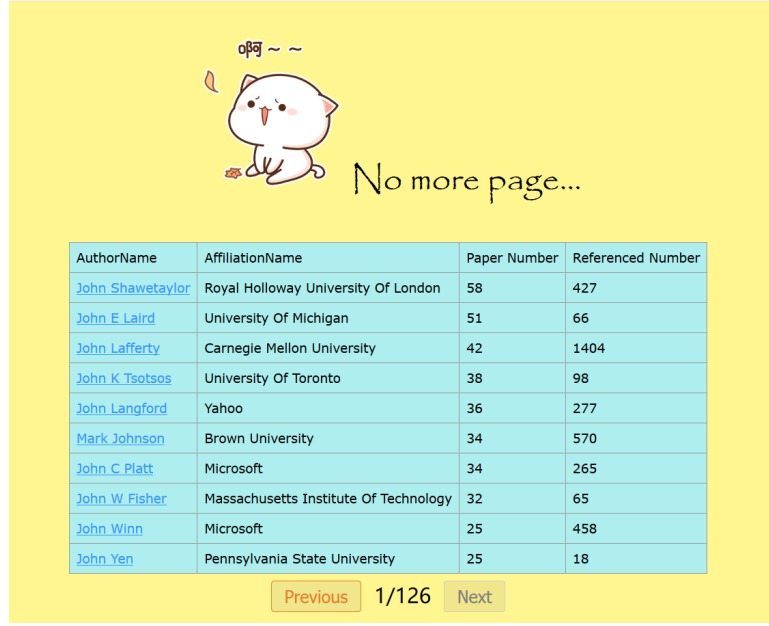


Figure 16: 3.6.3

## 4 Database Designed

A good database is the start of a good website. Our database mainly based on which in previous experiment, but we do some modify to boost it and improve the performance of our website.

### 4.1 Citation

Considering that citation times is important for readers to value an author and a paper, we add two columns to authors and papers.

#### 4.1.1 Analysis

1. For table authors:

For every author, first we find all the papers he/her has written (defined as set S). For each element x in S, we find how many times the paper x has been cited (defined as Cx).

Then the authors' citation times:

$$\sum Cx$$

2. For table papers:

For every paper, just find how many times this paper has been cited. Simply use sql SELECT and COUNT(\*).

#### 4.1.2 Codes

```

1 import pymysql
2 db = pymysql.connect(host='localhost',
3                       user='root',
4                       password='',
5                       db='main_db',
6                       charset='utf8',
7                       port=3306,
8                       cursorclass=pymysql.cursors.Cursor)
9 cursor=db.cursor()
10 #this function returns how many times this paper has been cited.
11 def cited (paperid):
```

```

12  sql1="""SELECT COUNT(*) FROM paper_reference where ReferenceID="{0}"
13  """.format(paperid)
14  cursor.execute(sql1)
15  results = cursor.fetchall()
16  return results[0][0]
17 def Allcited(authorid):
18     all=0
19     sql1 = """SELECT PaperID from paper_author_affiliation
20     WHERE AuthorID="{0}" """.format(authorid)
21     cursor.execute(sql1)
22     results =cursor.fetchall()
23     for row in results:
24         all+=cited(row[0])
25     return all
26
27 sql2="SELECT AuthorID from authors"
28 cursor.execute(sql2)
29 results2=cursor.fetchall()
30 for row in results2 :
31
32     sql = """UPDATE authors SET AllCited="{0}" WHERE AuthorID="{0}" """.format(Allcited(row[0]), row[0])
33     try:
34         cursor.execute(sql)
35         db.commit()
36     except:
37         db.rollback()
38
39 db.close()

```

#### 4.1.3 Result

- The author table:

| AuthorID | AuthorName                     | AllCited |
|----------|--------------------------------|----------|
| 000023A1 | markus grasmair                | 4        |
| 0000E2F9 | lampros tsinas                 | 0        |
| 000366AD | nazli goharian                 | 1        |
| 0003A3E4 | peter szeredi                  | 4        |
| 000548E1 | jacques chassin de kergommeaux | 1        |
| 0006031D | geoffrey oxholm                | 8        |
| 0006BF12 | babak nadjar araabi            | 0        |

- The papers table:

| PaperID  | Title                     | PaperPublishY | ConferenceID | Cited |
|----------|---------------------------|---------------|--------------|-------|
| 00002E77 | speech training sy 1979   | 47C39427      |              | 0     |
| 00003F20 | logical derivation c 1984 | 47C39427      |              | 0     |
| 000073E1 | proceedings of the 1983   | 43FD776C      |              | 0     |
| 00009805 | automated object i 2008   | 45701BF3      |              | 0     |
| 0000E395 | a general semantic 1981   | 47C39427      |              | 0     |
| 0000EB9E | intelligent printing 2009 | 45083D2F      |              | 0     |
| 0000F1CF | creating image ba: 1997   | 45083D2F      |              | 0     |
| 0000F908 | computational cha 1997    | 47C39427      |              | 3     |

- The page:

Basic information:

| Author Name      | Affiliation Name                    | Paper Number | Referenced Number |
|------------------|-------------------------------------|--------------|-------------------|
| John Shawetaylor | Royal Holloway University Of London | 58           | 427               |

Papers:

| Paper Name                                                              | Conference Name      | Paper Publish Year | Referenced Number | Main Author                         | Sequence of Searched Author |
|-------------------------------------------------------------------------|----------------------|--------------------|-------------------|-------------------------------------|-----------------------------|
| <a href="#">Pac Bayes Margins</a>                                       | <a href="#">NIPS</a> | 2003               | 13                | <a href="#">John Langford</a>       | 2                           |
| <a href="#">Two View Learning Svm 2k Theory And Practice</a>            | <a href="#">NIPS</a> | 2005               | 12                | <a href="#">Jason Farquhar</a>      | 4                           |
| <a href="#">Latent Semantic Kernels</a>                                 | <a href="#">ICML</a> | 2002               | 9                 | <a href="#">Nello Cristianini</a>   | 2                           |
| <a href="#">Optimizing Classifiers For Imbalanced Training Sets</a>     | <a href="#">NIPS</a> | 1999               | 7                 | <a href="#">Grigoris Karakoulas</a> | 2                           |
| <a href="#">A Column Generation Algorithm For Boosting</a>              | <a href="#">ICML</a> | 2000               | 7                 | <a href="#">Kristin P Bennet</a>    | 3                           |
| <a href="#">Dynamically Adapting Kernels In Support Vector Machines</a> | <a href="#">NIPS</a> | 1999               | 5                 | <a href="#">Nello Cristianini</a>   | 3                           |
| <a href="#">Spectral Kernel Methods For Clustering</a>                  | <a href="#">NIPS</a> | 2002               | 5                 | <a href="#">Nello Cristianini</a>   | 2                           |
| <a href="#">Sv Estimation Of A Distribution S Support</a>               | <a href="#">NIPS</a> | 2000               | 4                 | <a href="#">Bernhard Scholkopf</a>  | 4                           |
| <a href="#">Learning With The Set Covering Machine</a>                  | <a href="#">ICML</a> | 2001               | 4                 | <a href="#">Mario Marchand</a>      | 2                           |
| <a href="#">The Decision List Machine</a>                               | <a href="#">NIPS</a> | 2002               | 4                 | <a href="#">Marina Sokolova</a>     | 4                           |

## 4.2 SPEED-CACHE

### 4.2.1 To speed up

We find that for every search results, if the matched results are many, there will be more than 2 pages and it will take a time to turn to next page. So we use cache to speed up the page turning.

### 4.2.2 Codes

```

1 public function clear_all_cache()
2 {
3     $this->db->query("reset query cache");
4     return;
5 }
6
7 public function start_cache()
8 {
9     $this->db->query("set @@global.query_cache_type=on");
10    return;
11 }
```

## 5 Beautification

We write all of the .php in C:/xampp/htdocs/application/views/search, and .css in corresponding C:/xampp/htdocs/css/correspondig\_name\_of\_php.

### 5.1 Design for home

**title(<h1>)** We use < center >< /center > to make the title locates in the center of page, and add a picture which is aligned at the content's bottom and we limit its size.

```

1 <div class="alsp">
2     <center><h1>ScholarsSearch</h1></center>
4 </div>
```

We also give the title a class called alsp, and we set the whole class's style in home.css, where we change its font-size and font-family.

```

1 .alsp{
2     font-family:"chiller";
3     font-size:40px;
4 }
```



Figure 17: 5.1-1

**body** we set the body's style in home.css, which is aim to make all the page covered by yellow.

```
1 body{
2   background-repeat:no-repeat;
3   background-size:cover;
4   background-color:#FFF68F;
5 }
```

---

**input box** in home.php, we change the font-family of it, and make it align in center.

```
1 <form action="result" method="get" name="input">
2   <center><input class="InputContent" type="text_content" id="scholarname" name="scholarname"
3     style="font-family:Bradley Hand ITC"></center>
4 </form>
```

---

in home.css,we change its size.

```
1 input[type='text_content'] {
2   width: 600px;
3   height: 50px;
4   border-radius: 5px;
5   font-size:35px;
6
7 }
```

---

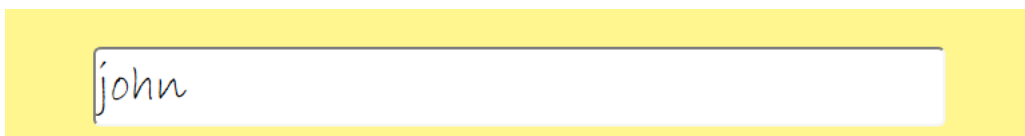


Figure 18: 5.1-2

**button** First, we need to set the integral style of button in home.css.

```
1 button{
2   font-family: 'Jokerman';
3   font-size: 15px;
4   line-height: 1;
5   color: #fff;
6   letter-spacing: 0.025em;
7   float:center;box-sizing:border-box;
8   background: #33CCFF;
9   padding: 18px 0 11px;
10  cursor: pointer;
11  border: 0;
12  border-radius: 15px;
13  min-width: 160px;
14  overflow: hidden;
```

```

15  -webkit-transform: translate(-50%, -50%);
16      transform: translate(-50%, -50%);
17  }

```

---



Figure 19: 5.1-3

Then , we add animation effect to it.

```

1  button span {
2      display: block;
3      position: relative;
4      z-index: 10;
5  }
6
7  button:after,
8  button:before{
9      padding: 18px 0 11px;
10     content: '';
11     position: absolute;
12     top: 0;
13     left: calc(-100% - 30px);
14     height: calc(100% - 29px);
15     width: calc(100% + 20px);
16     color: #7FFFD4;
17     border-radius: 2px;
18     -webkit-transform: skew(-25deg);
19     transform: skew(-25deg);
20 }
21
22 button:after {
23     background: #fff;
24     -webkit-transition: left 0.8s cubic-bezier(0.86, 0, 0.07, 1) 0.2s;
25     transition: left 0.8s cubic-bezier(0.86, 0, 0.07, 1) 0.2s;
26     z-index: 0;
27     opacity: 0.8;
28 }
29
30 button:before {
31     background: #13c276;
32     z-index: 5;
33     -webkit-transition: left 1s cubic-bezier(0.86, 0, 0.07, 1);
34     transition: left 1s cubic-bezier(0.86, 0, 0.07, 1);
35 }
36
37 button:hover:after {
38     left: calc(0% - 10px);
39     -webkit-transition: left 0.8s cubic-bezier(0.86, 0, 0.07, 1);
40     transition: left 0.8s cubic-bezier(0.86, 0, 0.07, 1);
41 }
42
43 button:hover:before {
44     left: calc(0% - 10px);
45     -webkit-transition: left 1s cubic-bezier(0.86, 0, 0.07, 1);
46     transition: left 1s cubic-bezier(0.86, 0, 0.07, 1);
47 }

```

---



Figure 20: 5.1-4

**Loading Animation** Since search may cause some time, we add a loading animation to our page. We use ionic to draw the animation and use its default .css, and place a gif and some words on it too.

```

1 <div id="loader">
2   <div class="row" ng-app="ionicApp" id="spinnerDiv">
3     <div class="col-lg-6 col-lg-offset-3">
4       <p style="width:15%; margin: auto ;padding-top:10%"><!-- -->
5         <ion-spinner icon="bubbles" style="width:10%;stroke:#33cd5f;fill:#33cd5f;"></ion-spinner>
6       </p>
7     </div>
8   </div>
9   <p id='p1'>Loading....</p>
10 </div>
11 <script type="text/javascript" src="/js/ionic.bundle.min.js"></script>

```

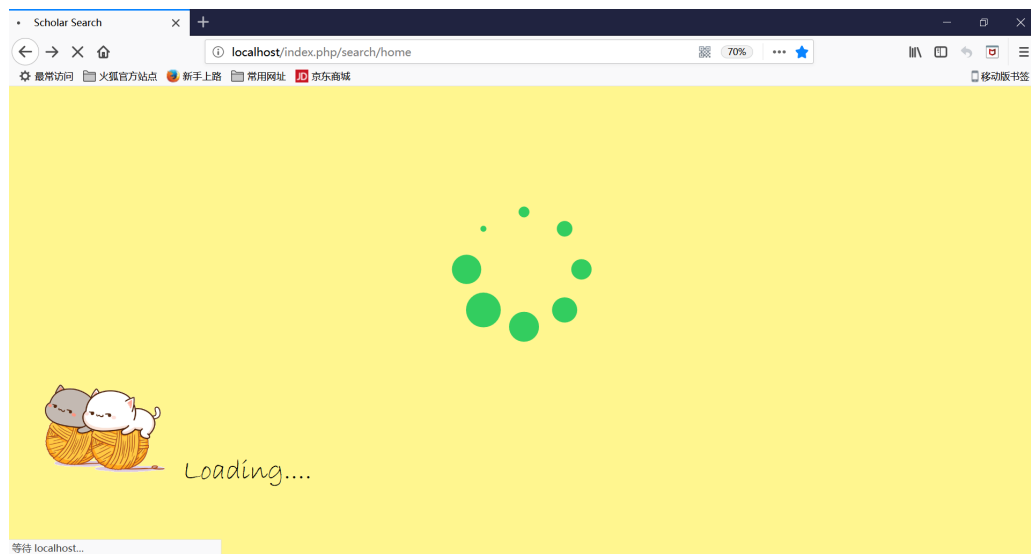


Figure 21: 5.1-5

**Link To Producer** We place a gif and some words at bottom left, and add hyperlink to it.

```

1 <p id='p2'><a href="producer" ></a>Click me to
  see the staff!</p>

```



Figure 22: 5.1-6

**Other** To make the page more vivid, we use canvas to draw some bubble hovering the button.

```

1  canvas{
2    width: 420px;
3    height: 420px;
4    background: transparent;
5    position: absolute;
6    top: 50%;
7    left: 50%;
8    -webkit-transform: translate(-50%, -50%);
9    transform: translate(-50%, -50%);
10   pointer-events: none;
11 }
12 <script type="text/javascript" src="/js/home/circle.js"></script>

```

---

Then we can use `< canvas >` `< /canvas >` to place it.

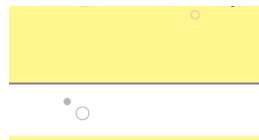


Figure 23: 5.1-7

### Problem met in task

1. since we have 3 buttons,we need to keep them in a line.

(a) So we use css list to set these buttons.

```

1  .foot{width: 100%; height: 40px; text-align: center;}
2  .foot ul{display: inline; margin-left: 30px;}
3  .foot ul li{display: inline-block; margin-left: 100px; line-height: 50px;width: 13%;}

```

---

So in home.php, we write the button as:

```

1  <div class="foot">
2    <ul>
3      <li>
4        <button type="submit" onclick="author()"><span>Search Name</span></button>
5      </li>
6      <li>
7        <button type="submit" onclick="paper()"><span>Search Title</span></button>
8      </li>
9      <li>
10       <button type="submit" onclick="confer()"><span>Search Conference</span></button>
11     </li>
12   </ul>
13 </div>

```

---

2. though we know how to draw a loading animation, we cannot show it in correct time.

- (a) To implement the loading effect, we write some function in js. When the page has loaded, the loader will disappear and when click the button which means you trigger onclick(), the body will disappear and loader will appear.

```
1  window.onload = pageLoad;
2
3  function pageLoad()
4  {
5      document.getElementById("loader").style.display = "none";
6  }
7
8
9  function bubble()
10 {
11     document.getElementById("loader").style.display = "inline";
12     document.getElementById("mybd").style.display = "none";
13 }
14
15 function author()
16 {
17     bubble();
18     document.input.action="result";
19 }
20
21 function paper()
22 {
23     bubble();
24     document.input.action="title";
25 }
26
27 function confer()
28 {
29     bubble();
30     document.input.action="conferenceresult";
31 }
32 angular.module('ionicApp', ['ionic'])
```

---

3. if you put the picture just between the text, you will find it just align in the center, which is ugly.

- (a) we set the style of picture that is: align :button.

## 5.2 Design For Producer

In this html, we use bootstrap.min.css and font-awesome.css to make the basic format of the page. And we write `<styletype = "text/css">` directly in producer.html, to set our own format.

```
1  <link href="http://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">
2  <link href="http://cdn.bootcss.com/font-awesome/4.6.3/css/font-awesome.min.css" rel="stylesheet">
3  <style type="text/css">
4  .demo{
5      padding: 2em 0;
6  }
7  .our-team{
8      overflow: hidden;
9      border-radius: 50%;
10     position: relative;
11 }
12 .our-team img{
13     width: 100%;
14     height: auto;
15     position: static;
16     right: 0;
17     transition: all 0.4s ease-in-out 0s;
18 }
19 .our-team:hover img{
20     right: 60%;
21 }
22 .our-team .team-content{
23     width: 80%;
```

```

24     height: 100%;
25     background: #f1f1f1;
26     padding: 20% 30px 0;
27     position: absolute;
28     top: 0;
29     right: -80%;
30     transition: all 0.4s ease-in-out 0s;
31 }
32 .our-team:hover .team-content{
33     right: 0;
34 }
35 .our-team .title{
36     font-size: 40px;
37     color: #000000;
38     margin: 0 0 10px 0;
39 }
40 .our-team .post{
41     display: block;
42     font-size: 15px;
43     font-weight: 900;
44     color: #333;
45     text-transform: uppercase;
46     margin-bottom: 20px;
47 }
48 </style>

```

---

Then we just add our staff's basic information into it.

```

1  <div class="demo">
2    <div class="container">
3      <div class="row">
4        <div class="col-md-3 ">
5          <div class="our-team">
6            
7            <div class="team-content">
8              <h3 class="title"> < /h3>
9              <span class="post">Back-end Worker And debugger</span>
10
11            </div>
12          </div>
13        </div>
14      </div>
15
16      <div class="col-md-3">
17        <div class="our-team">
18          
19          <div class="team-content">
20            <h3 class="title"> < /h3>
21            <span class="post">Web Designer<span>
22
23          </div>
24        </div>
25      </div>
26
27      <div class="col-md-3 ">
28        <div class="our-team">
29          
30          <div class="team-content">
31            <h3 class="title"> < /h3>
32            <span class="post">Database Developer</span>
33
34          </div>
35        </div>
36      </div>
37
38      <div class="col-md-3 ">
39        <div class="our-team">
40          
41          <div class="team-content">
42            <h3 class="title"> < /h3>
43            <span class="post">Autocomplete Developer</span>
44
45          </div>

```

```

46     </div>
47 </div>
48 </div>
49 </div>
50 </div>

```

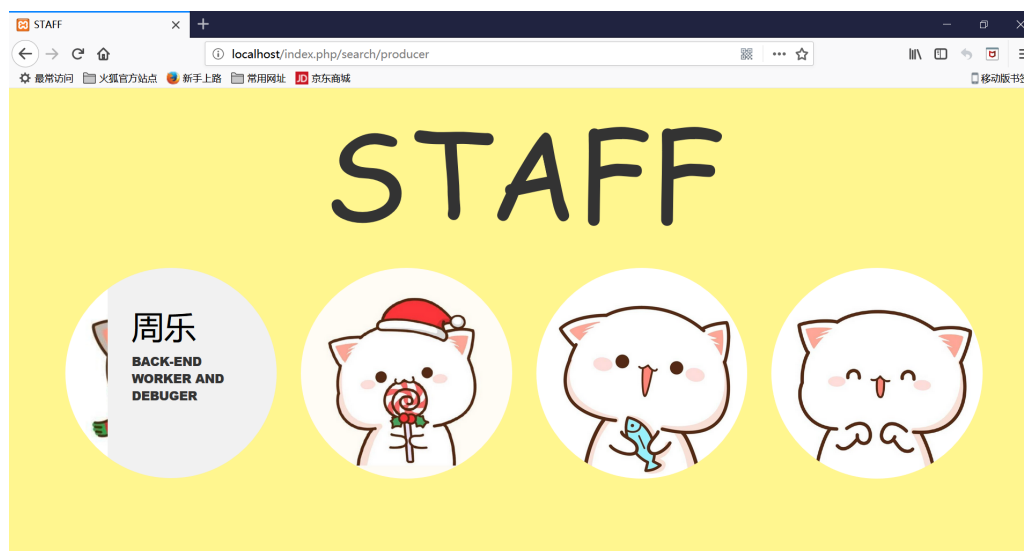


Figure 24: 5.2

## Problem met in task

1. how to implement that click on the picture and jump to he page?
  - (a) we can also add hyperlinks to picture! or use onclick().

## 5.3 Design for Result

In result.php,except for the basic format write in home, we need to add some other format of the item.

**table** In result.php, we use table to show the data.Make it in center and when mouse point on it and leave it it will change its color.

```

1     <center>
2         <table border="1" id="tb1" class="hovertable">
3             <öphp
4             echo "<tr><td>AuthorName</td><td>AffiliationName</td></tr>";
5             for ($i=0;$i<$datanum;$i++)
6             { $suppro=ucwords($data[$i][0]['AuthorName']);
7               $supaffi=ucwords($data[$i][1]['AffiliationName']);
8               echo "<tr onmouseover=\"this.style.backgroundColor='#87CEEB';\"
9                 onmouseout=\"this.style.backgroundColor='#AFEEEE';\">
10                 <td><a onclick=\"bubble()\" href='authoröauthorid={$data[$i][0]['AuthorID']}\"
11                   >{$suppro}</a></td>
12                 <td>{$supaffi}</td>
13                 </tr>";
14             }
15             </table>
16         </center>

```

And in result.css, we need to set the basic format of it, including each item of table's color and hyperlink's property.

```

1 table.hovertable {
2     font-family: verdana,arial,sans-serif;

```

```

3   color:#000000;
4   border-width: 1px;
5   border-color: #999999;
6   border-collapse: collapse;
7   font-size:15px;
8 }
9
10 table.hovertable th {
11     background-color:#AFEEEE;
12     border-width: 3px;
13     padding: 8px;
14     border-style: solid;
15     border-color: #a9c6c9;
16 }
17
18 table.hovertable tr {
19     background-color:#AFEEEE;
20 }
21
22 table.hovertable td {
23     border-width: 1px;
24     padding: 8px;
25     border-style: solid;
26     border-color: #999999;
27 }
28 }
29
30 a:link{
31     color:#3366FF;
32     text-decoration:underline;
33 }
34 a:visited{
35     color:#3399FF;
36     text-decoration:none;
37 }
38 a:hover{
39     color:#000000;
40     text-decoration:none;
41 }
42 a:active{
43     color:#FFFFFF;
44     text-decoration:none;
45 }

```

---

| AuthorName                        | AffiliationName                     | Paper Number | Referenced Number |
|-----------------------------------|-------------------------------------|--------------|-------------------|
| <a href="#">Michael I Jordan</a>  | University Of California Berkeley   | 162          | 2723              |
| Thorsten Joachims                 | Cornell University                  | 74           | 2019              |
| <a href="#">Joseph Y Halpern</a>  | Ibm                                 | 59           | 96                |
| <a href="#">John Shawetaylor</a>  | Royal Holloway University Of London | 58           | 427               |
| <a href="#">Aravind K Joshi</a>   | University Of Pennsylvania          | 58           | 176               |
| <a href="#">Joachim M Buhmann</a> | Eth Zurich                          | 58           | 176               |

Figure 25: 5.3-1

**title** To beautify the page , we add a title of it.

```

1  <center><p id="p1">We find these!</p></center>

```

---



Figure 26: 5.3-2

**Page Button** Since in this page, we have add page-turning function, so we need to change the style of buttons.

```

1      <center>
2          <button id="bt1" onclick="previouspage()" class="btn btn-orange
              btn-border-o">Previous</button>
3          <span id='pg'>1</span>/<öphp echo ceil($alldatanum/10);ö>
4          <button id="bt2" onclick="nextpage()" class="btn btn-orange btn-border-o">Next</button>
5      </center>

```

We have set their class, then we should write detailed class setting in authorresult.css.

```

1  /* Reset */
2  .animate,
3  .btn,
4  .btn-border-o:before,
5  .btn-border-o:after,
6  .btn-border:before,
7  .btn-border:after,
8  .btn-border-rev-o:before,
9  .btn-border-rev-o:after,
10 .btn-border-rev:before,
11 .btn-border-rev:after,
12 .btn-fill-vert-o:before,
13 .btn-fill-vert-o:after,
14 .btn-fill-vert:before,
15 .btn-fill-vert:after,
16 .btn-fill-horz-o:before,
17 .btn-fill-horz-o:after,
18 .btn-fill-horz:before,
19 .btn-fill-horz:after{
20     -webkit-transition: all 0.3s;
21     transition: all 0.3s;
22 }
23
24 .btn{
25     display: inline-block;
26     line-height: 35px;
27     margin: 8px;
28     padding: 0 15px;
29     font-size: 20px;
30     position: relative;
31     opacity: .999;
32     border-radius: 3px;
33 }
34
35 .btn-border-o{
36     background-color: #F0E68C;
37     border: 1px solid #d0d0d0;
38     color: #808080;
39 }
40
41 .btn-border-o:before,
42 .btn-border-o:after{
43     content: '';

```

```

44     border-style: solid;
45     position: absolute;
46     z-index: 5;
47     border-radius: 3px;
48     box-sizing: content-box;
49 }
50
51 .btn-border-o:before{
52     width: 0;
53     height: 100%;
54     border-width: 1px 0 1px 0;
55     top: -1px;
56     left: 0;
57     -webkit-transition-delay: 0.05s;
58     transition-delay: 0.05s;
59 }
60
61 .btn-border-o:after {
62     width: 100%;
63     height: 0;
64     border-width: 0 1px 0 1px;
65     top: 0;
66     left: -1px;
67 }
68
69 .btn-border-o:hover:before {
70     width: 100%;
71 }
72
73 .btn-border-o:hover:after {
74     height: 100%;
75 }
76
77 .btn-border-o.btn-orange:before, .btn-border-o.btn-orange:after {
78     border-color: #e67e22;
79 }
80
81 .btn-border-o.btn-orange:hover {
82     color: #e67e22;
83 }

```

---

Figure 27: 5.2-3

**No More Page Warning** since the users may be false operated that he/she will try to page an inexistent page, o it's time to make a warning.

```

1     <div id='cata'>
2         <center><p id='p2'>No
3         more page...</p></center>
4     </div>

```

---

We have place the picture, and wrap it into a class. To realize the function, we need to use some js ways. First,we write two function to make the switch between title and the warning.

```

1  function cat_a()
2  {
3      document.getElementById("p1").style.display = "none";
4      document.getElementById("cata").style.display = "block";
5  }
6
7  function cat_disappear()
8  {
9      document.getElementById("p1").style.display = "block";
10     document.getElementById("cata").style.display = "none";
11 }

```

---

Then we need to bind the function with button. At the same time, we add some other property of it, that is when the warning come out, if you then press the true button, or do nothing but wait for 1000ms, the warning will disappear.

```

1  function nextpage()
2  {
3      if(page< <?php echo ceil($alldatanum/10);?>)
4      {
5          page=page+1;
6          document.getElementById("pg").innerHTML="<span id='pg'>"+page+"</span>";
7          loadXMLDoc();
8          cat_disappear();
9      }
10     else
11     {
12         cat_a();
13         setTimeout("cat_disappear()", "1000");
14     }
15 }
16
17 }
18
19 function previouspage()
20 {
21     if(page!=1)
22     {
23         page=page-1;
24         document.getElementById("pg").innerHTML="<span id='pg'>"+page+"</span>";
25         loadXMLDoc();
26         document.getElementById("cata").style.display = "none";
27     }
28     else
29     {
30         cat_a();
31         setTimeout("cat_disappear()", "2000");
32     }
33 }
34 }

```

---



Figure 28: 5.3-4

**Home Button** We use another type of button to make home button. First, we create wrap it with hyperlink, and wrap it with a class.

```

1  <a href='home'>
2  <button class="bubbly-button">Home</button>
3  </a>

```

---

Then, we need to write detailed design for bubbly-button. While we click on it, there will be bubbles hover it.

```

1  .bubbly-button {
2      font-family: Comic Sans MS;
3      display: inline-block;
4      font-size: 1em;
5      padding: 1em 2em;

```

```

6     margin-top: 100px;
7     margin-bottom: 60px;
8     -webkit-appearance: none;
9     appearance: none;
10    background-color: #ff0081;
11    color: #fff;
12    border-radius: 4px;
13    border: none;
14    cursor: pointer;
15    position: absolute;
16    left: 0px;
17
18    transition: transform ease-in 0.1s, box-shadow ease-in 0.25s;
19    box-shadow: 0 2px 25px rgba(255, 0, 130, 0.5);
20 }
21
22 .bubbly-button:focus {
23     outline: 0;
24 }
25
26 .bubbly-button:before, .bubbly-button:after {
27     position: absolute;
28     content: '';
29     display: block;
30     width: 140%;
31     height: 100%;
32     left: -20%;
33     z-index: -1000;
34     transition: all ease-in-out 0.5s;
35     background-repeat: no-repeat;
36 }
37
38 .bubbly-button:before {
39     display: none;
40     top: -75%;
41     background-image:
42         radial-gradient(circle, #ff0081 20%, transparent 20%),
43         radial-gradient(circle, transparent 20%, #ff0081 20%, transparent 30%),
44         radial-gradient(circle, #ff0081 20%, transparent 20%),
45         radial-gradient(circle, #ff0081 20%, transparent 20%),
46         radial-gradient(circle, transparent 10%, #ff0081 15%, transparent 20%),
47         radial-gradient(circle, #ff0081 20%, transparent 20%),
48         radial-gradient(circle, #ff0081 20%, transparent 20%),
49         radial-gradient(circle, #ff0081 20%, transparent 20%),
50         radial-gradient(circle, #ff0081 20%, transparent 20%);
51     background-size: 10% 10%, 20% 20%, 15% 15%, 20% 20%, 18% 18%, 10% 10%, 15% 15%, 10% 10%, 18% 18%;
52 }
53
54 .bubbly-button:after {
55     display: none;
56     bottom: -75%;
57     background-image:
58         radial-gradient(circle, #ff0081 20%, transparent 20%),
59         radial-gradient(circle, #ff0081 20%, transparent 20%),
60         radial-gradient(circle, transparent 10%, #ff0081 15%, transparent 20%),
61         radial-gradient(circle, #ff0081 20%, transparent 20%),
62         radial-gradient(circle, #ff0081 20%, transparent 20%),
63         radial-gradient(circle, #ff0081 20%, transparent 20%),
64         radial-gradient(circle, #ff0081 20%, transparent 20%);
65     background-size: 15% 15%, 20% 20%, 18% 18%, 20% 20%, 15% 15%, 10% 10%, 20% 20%;
66 }
67
68 .bubbly-button:active {
69     transform: scale(0.9);
70     background-color: #e60074;
71     box-shadow: 0 2px 25px rgba(255, 0, 130, 0.2);
72 }
73
74 .bubbly-button.animate:before {
75     display: block;
76     animation: topBubbles ease-in-out 0.75s forwards;
77 }
78

```

```

79 .bubbly-button.animate:after {
80     display: block;
81     animation: bottomBubbles ease-in-out 0.75s forwards;
82 }
83
84 @keyframes topBubbles {
85     0% {
86         background-position: 5% 90%, 10% 90%, 10% 90%, 15% 90%, 25% 90%, 25% 90%, 40% 90%, 55% 90%, 70%
            90%;
87     }
88     50% {
89         background-position: 0% 80%, 0% 20%, 10% 40%, 20% 0%, 30% 30%, 22% 50%, 50% 50%, 65% 20%, 90% 30%;
90     }
91     100% {
92         background-position: 0% 70%, 0% 10%, 10% 30%, 20% -10%, 30% 20%, 22% 40%, 50% 40%, 65% 10%, 90%
            20%;
93         background-size: 0% 0%, 0% 0%, 0% 0%, 0% 0%, 0% 0%, 0% 0%, 0% 0%;
94     }
95 }
96
97 @keyframes bottomBubbles {
98     0% {
99         background-position: 10% -10%, 30% 10%, 55% -10%, 70% -10%, 85% -10%, 70% -10%, 70% 0%;
100    }
101    50% {
102        background-position: 0% 80%, 20% 80%, 45% 60%, 60% 100%, 75% 70%, 95% 60%, 105% 0%;
103    }
104    100% {
105        background-position: 0% 90%, 20% 90%, 45% 70%, 60% 110%, 75% 80%, 95% 70%, 110% 10%;
106        background-size: 0% 0%, 0% 0%, 0% 0%, 0% 0%, 0% 0%, 0% 0%, 0% 0%;
107    }
108 }

```

---

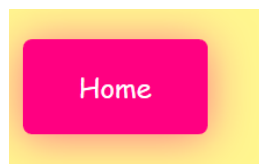


Figure 29: 5.3-5

### Problem met in task

1. there are two kinds of button, but there all belongs to the button class, so one format will cover another.
  - (a) by looking up the materials, I find that we can change the class even if it's the reserved class. So I change one kind of button's class into btn.
2. while change the color of table, the hyperlink become less visiable.
  - (a) there is also css property to set the color of hyperlink, so we set the color for our convenience.
3. when the cat disappear and appear , there will surplus two jbṙ.
  - (a) change style="inline" to style="block".

## 5.4 Author

**Force graph** To add readability, we can set about making legend of force graph. In this way, we use svg to draw circle and place the text around it.

```

1     <svg width="80" height="40">
2         <circle cx="35" cy="25" r="5" fill="#333399" />
3         <text x="5" y="20" style="fill:black;">Current
4     </svg>
5     <svg width="80" height="40">

```

```

6      <circle cx="35" cy="25" r="5" fill="#ff3366" />
7      <text x="5" y="20" style="fill:black;">Student
8  </svg>
9  <svg width="80" height="40">
10     <circle cx="35" cy="25" r="5" fill="#66ccff" />
11     <text x="5" y="20" style="fill:black;">Teacher
12  </svg>
13  <svg width="80" height="40">
14     <circle cx="35" cy="25" r="5" fill="#ffcc99" />
15     <text x="5" y="20" style="fill:black;">Inrelevant
16  </svg>

```

---

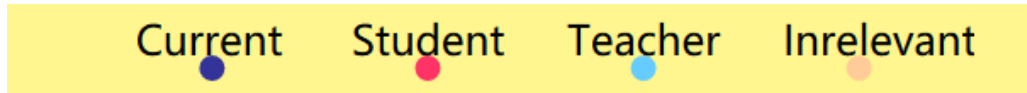


Figure 30: 5.4-1

At last, we need to write some functions to let it appear and disappear. And we bind these functions to two buttons whose class are bubbly-button.

```

1  <button class="bubbly-button" id="bubbly-button-fa" onclick="force_appear()">Force Graph</button>
2  <button class="bubbly-button" id="bubbly-button-fd" onclick="force_disappear()">Disappear</button>
3
4  function force_a()
5  {
6      document.getElementById("force_graph").style.display = "inline";
7      document.getElementById("bubbly-button-fa").style.display = "none";
8      document.getElementById("bubbly-button-fd").style.display = "block";
9  }
10
11 function force_appear()
12 {
13     setTimeout("force_a()", "500");
14     setTimeout("document.getElementById('p1').style.display = 'none'", "500");
15 }
16
17 function force_d()
18 {
19     document.getElementById("bubbly-button-fd").style.display = "none";
20     document.getElementById("bubbly-button-fa").style.display = "block";
21     document.getElementById("force_graph").style.display = "none";
22 }
23 function force_disappear()
24 {
25     setTimeout("force_d()", "500");
26     setTimeout("document.getElementById('p1').style.display = 'block'", "500");
27 }

```

---



Figure 31: 5.4-2

**Tree Graph** Since two svg graph in the same page will interact each other, we use a button to jump to another page: tree.php.

```

1  <a href='tree.php?authorid=<?php echo $authorid;?>'>
2  <button class="bubbly-button">Tree Graph</button>
3  </a>

```

---

In tree graph, we change the form of node into rounded rectangle, and set the filled color.

```
1 .node rect {
2   fill: #fff;
3   stroke: steelblue;
4   stroke-width: 1px;
5 }
6 nodeEnter.append("rect")
7   .attr("x", "-80")
8   .attr("rx", "20")
9   .attr("ry", "20")
10  .attr("width", 1e-6)
11  .attr("height", 1e-6)
12  .style("fill", function(d) { return d._children & "#ccccff" : "#fff"; });
```

---

Also, we need to fix the size of svg container.

```
1 <div id='tree_graph'>
2 <svg width="1600" height="1600"></svg>
3 </div>
4 var margin = {top: 40, right: 120, bottom: 20, left: 700},
5 width = 1600,
6 height = 1600;
```

---

And then add a button called disappear to jump back.

```
1 <a href='author&authorid=<?php echo $authorid;>'><button
   class="bubbly-button">Disappear</button></a>
```

---

## Problem met in task

1. since the color of nodes in force graph is random, we change some function to fix the color.

```
1 var node = svg_force.append("g")
2   .attr("class", "nodes")
3   .selectAll("circle")
4   .data(nodes)
5   .enter().append("circle")
6   .attr("r", 5)
7   .attr("fill", function(d) {
8     if(d.group==1) return "#333399";
9     if(d.group==2) return "#ff3366";
10    if(d.group==3) return "#66ccff";
11    if(d.group==4) return "#ffcc99";})
12   .call(d3.drag()
13     .on("start", dragstarted)
14     .on("drag", dragged)
15     .on("end", dragended));
```

---

2. since two graph are drawn by svg, they have duplicate name , and they will affect each other.

- (a) we move the tree graph to a new page, when click on the button, it will jump to another page called tree.php.

## 5.5 404 page

Considering the user may enter some wrong name that not belong to our database, we set a 404 page to solve this situation.

We use 404.css to make animation, the module is downloaded from Internet (We just change its color). Then in 404.html, we use the css format like:

```
1 <!DOCTYPE html>
2 <html lang="zh">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="css/404.css" />
7
```

```

8 </head>
9 <body>
10 <nav class="shelf">
11 <a class="book home-page" href="home">Home</a>
12 <a class="book about-us" href="producer">Producer</a>
13 <span class="book not-found"></span>
14 <span class="door left"></span>
15 <span class="door right"></span>
16 </nav>
17 <h1>Error</h1>
18 <p>We cannot find it...</p>
19
20 </body>
21 </html>

```

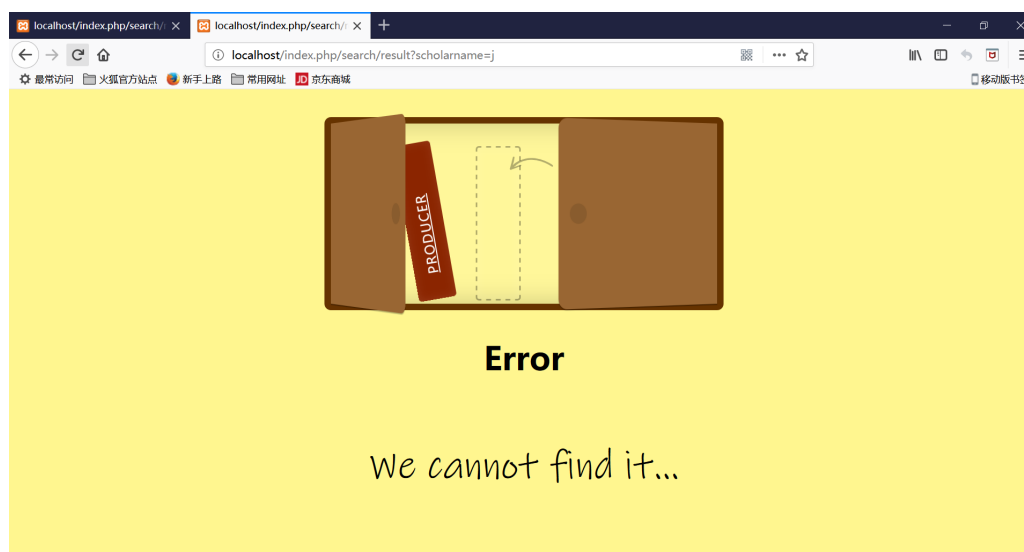


Figure 32: 5.5

## 6 Optimization

### 6.1 Introduction

Since all the features and functions are built successfully, somewhere such as force-directed graph show bad performance when loading some famous authors who have more than 100 students. Besides, the codes of our website are too complicated and chaos, which is difficult for us to update. Thus, we do some work in simplify our code and decrease the amount of calculating in some process. Here, we follow these thought to optimize:

- Decrease usage of SQL statement
- When faced  $O(n^2)$ , reduce scale of  $n$
- Using same SQL codes to use cache while paging or searching

### 6.2 An example of running speed Optimization

At first, the part of code in author.php which is used to find every two author's relationship likes this:

```

1 for($i=0;$i<count($subauthorid);++$i)
2 for($j=$i+1;$j<count($subauthorid);++$j)
3 if(count($this->search_model->get_furtherrelationship($subauthorid[$i],$subauthorid[$j])))
4 $links[]=array('source'=>$subauthorid[$i],'target'=>$subauthorid[$j],'value'=>1);

```

We can find it that the complex of it is  $O(n^2)$ , but no matter how we modify, it can not avoid this type of complex. At last, we find that if we can get each author's all advisors and teachers by one scanning and store

them in a small scale array, then do the same process, we can get 7 times fast than origin, which means all authors page can be loaded in one second. The code after optimization like this:

```

1  foreach($subauthorid as $subid)
2  {
3      $subauthor[$subid] = $this->search_model->get_id_author($subid)[0]['AuthorName'];
4  }
5
6  foreach($subauthorid as $subid)
7  {
8      $links[]=array('source'=>$id,'target'=>$subid,'value'=>1);
9      $query6 = $this->search_model->get_rela2($subid);
10     foreach($query6 as $row)
11     if(in_array($row['StudentID'],$subauthorid))
12     $links[]=array('source'=>$subid,'target'=>$row['StudentID'],'value'=>1);
13 }

```

---

### 6.3 An example of code Optimization

To make sure codes in front-end are correctly and efficiently loaded, we sequence the code as follow:

1. `<html>`
2. `<head>`
3. `<link>` (public css script) `</link>`
4. `<style>` (private css script) `</style>`
5. `<title>` `</title>`
6. `</head>`
7. `<body>` (...) `</body>`
8. `<script>` (js script) `</script>`
9. `<script>`
10. define variables
11. define functions
12. process
13. `</script>`
14. `</html>`

For result page, we modify the code like this:

```

1  <html>
2
3
4
5  <head>
6      <link rel="stylesheet" type="text/css" href="/css/authorresult/authorresult.css">
7
8      <div id="loader">
9          <div class="row" ng-app="ionicApp" id="spinnerDiv">
10             <div class="col-lg-6 col-lg-offset-3">
11                 <p style="width:15%; margin: auto ;padding-top:10%"><!-- -->
12                     <ion-spinner icon="bubbles"
13                         style="width:10%;stroke:#33cd5f;fill:#33cd5f;"></ion-spinner>
14                 </p>
15             </div>
16         </div>
17         <p id='p3'>Loading....</p>
18     </div>

```

```

19     <title><öphp echo "Scholar Search";ö></title>
20 </head>
21
22
23
24
25 <body bgcolor="#FFF68F">
26     <div id='mybd'>
27         <center><p id="p1">We find these!</p></center>
29         <div id='cata'>
30             <center><p id="p2">No
31                 more page...</p></center>
32         </div>
33         <font size=5>
34             <center>
35                 <table border="1" id="tb1" class="hovertable">
36                     <öphp
37                     echo "<tr><td>AuthorName</td><td>AffiliationName</td><td>Paper Number</td><td>Referenced
38                         Number</td></tr>";
39                     for ($i=0;$i<$datanum;$i++)
40                     { $suppro=ucwords($data[$i][0]['AuthorName']);
41                       $supaffi=ucwords($data[$i][1]['AffiliationName']);
42                       echo "<tr onmouseover=\"this.style.backgroundColor='#87CEEB';\"
43                           onmouseout=\"this.style.backgroundColor='#AFEEEE';\">
44                           <td><a onclick=\"bubble()\"
45                               href='authoröauthorid={$data[$i][0]['AuthorID']}'>{$suppro}</a></td>
46                           <td>{$supaffi}</td>
47                           <td>{$data[$i][0]['Papernum']}</td>
48                           <td>{$data[$i][0]['AllCited']}</td>
49                           </tr>";
50                     }
51                     ö>
52                 </table>
53             </center>
54             <center>
55                 <button id="bt1" onclick="previouspage()" class="btn btn-orange
56                     btn-border-o">Previous</button>
57                 <span id='pg'>1</span><öphp echo ceil($alldatanum/10);ö>
58                 <button id="bt2" onclick="nextpage()" class="btn btn-orange btn-border-o">Next</button>
59             </center>
60         </font>
61         <a href='home'>
62         <button class="bubbly-button">Home</button>
63         </a>
64     </div>
65 </body>
66
67
68
69
70
71
72 <script type="text/javascript" src="/js/jquery-1.8.2.min.js"></script>
73 <script type="text/javascript" src="/js/jquery.pricetable.min.js"></script>
74 <script type="text/javascript" src="/js/jquery.js"></script>
75 <script src="/js/ionic.bundle.min.js"></script>
76 <script>
77
78     var page = 1;
79
80     window.onload = pageLoad;
81
82     function nextpage()
83     {
84         if(page< <öphp echo ceil($alldatanum/10);ö>)
85         {
86             page=page+1;
87             document.getElementById("pg").innerHTML="<span id='pg'>"+page+"</span>";
88             loadXMLDoc();
89             cat_disappear();
90         }
91         else
92         {
93             cat_a();
94             setTimeout("cat_disappear()", "1000");
95         }
96     }
97

```

```

86     }
87
88 }
89
90 function previouspage()
91 {
92     if(page!=1)
93     {
94         page=page-1;
95         document.getElementById("pg").innerHTML="<span id='pg'>"+page+"</span>";
96         loadXMLDoc();
97         document.getElementById("cata").style.display = "none";
98     }
99     else
100     {
101         cat_a();
102         setTimeout("cat_disappear()", "2000");
103     }
104 }
105 }
106
107 function loadXMLDoc()
108 {
109     var xmlhttp;
110     if (window.XMLHttpRequest)
111     {
112         xmlhttp=new XMLHttpRequest();
113     }
114     else
115     {
116         xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
117     }
118     xmlhttp.onreadystatechange=function()
119     {
120         if (xmlhttp.readyState==4 && xmlhttp.status==200)
121         {
122             document.getElementById("tb1").innerHTML=xmlhttp.responseText;
123         }
124     }
125     xmlhttp.open("GET","newresultöscholarname=<öphp echo $scholarname;ö>&page="+page,true);
126     xmlhttp.send();
127 }
128
129 function cat_a()
130 {
131     document.getElementById("p1").style.display = "none";
132     document.getElementById("cata").style.display = "block";
133 }
134
135 function cat_disappear()
136 {
137     document.getElementById("p1").style.display = "block";
138     document.getElementById("cata").style.display = "none";
139 }
140
141 function pageLoad()
142 {
143     document.getElementById("loader").style.display = "none";
144     document.getElementById("cata").style.display = "none";
145     document.getElementById("p1").style.display = "block";
146 }
147
148 function bubble()
149 {
150     document.getElementById("loader").style.display = "block";
151     document.getElementById("mybd").style.display = "none";
152 }
153
154 angular.module('ionicApp', ['ionic'])
155
156 </script>
157
158

```

## 7 Conclusion

For the given tasks, our group realize all of them except use search engine. Even though paper recommend sometimes does a kind of "strict" and search speed does not so fast as some other group, our website has unique and distinct theme origin by our group. Besides, the performance of the website is fast enough to face such amount of data.

Thanks for your reading!