基于爬取到数据库的数据，实现数据展示的网站

（如果图片加载不出来，请连接外网vpn（因为图片是放到github项目中的），或者直接进入我的github项目查看images文件夹或直接看pdf版的实验报告）

下面是该项目的要求：（由于老师两个ppt中对大作业的要求并不一样，所以我取了两个要求的并集）

```
1、用户可注册登录网站，非注册用户不可登录查看数据
2、用户注册、登录、查询等操作记入数据库中的日志
3、爬虫数据查询结果列表支持分页和排序
4、用Echarts或者D3实现3个以上的数据分析图表展示在网站中
5、实现一个管理端界面，可以查看（查看用户的操作记录）和管理（停用启用）注册用户。
6、实现查询词支持布尔表达式 （比如“新冠 AND 肺炎”或者“新冠 OR 肺炎”）
```

由于本次实验的功能较多、代码量较大，老师提供了一份基础代码已经完成了上述项目的70%，而我目前由于准备考研没有太多时间，所以直接选择使用老师提供的基础代码，并在此基础之上对功能进行一定的优化与完善。

项目架构

本项目使用了express架构和Angular JS框架。

各项目文件的作用：

```
bin                                 启动文件
    --www
conf
    --mysqlConf.js                  存放mysql数据库的密码、以及要进行操作的数据库名称
dao                                 与数据库进行交互，很多sql语句
    --logDAO.js                     对user_action（日志）表格进行插入和查询
    --newsDAO.js                    对eastmoney（新闻报道）表格进行查询
    --regDAO.js                     对admin（管理员操作）、adminer（管理员名单）表格进行插入
与查询
    --userDAO.js                    对user（用户）表格进行插入与查询
public                              静态资源文件目录
    --images                        图片用于词云显示的样式
    --javascripts
        --dist                      存放词云所需样式
        --index.js
        --news.js                   存放angular.module模块，接收前端html数据，并将之处理后
传递给路由文件
    --stylesheets                   css样式
    --index.html                    登录注册前端页面
    --news.html                     包含导航栏、图表、管理员功能的前端页面
    --search.html                   搜索功能的前端页面（其实是news.html的一部分单独拿出来了）
routes                              路由文件，与dao文件夹的文件进行交互，以指定的http请求方式
暴露给用户，并在用户请求后将结果返回
    --news.js                       搜索、图表、管理员功能的路由
    --user.js                       注册登录的路由
    --wordcut.js                    分词功能、词云所需
    --freqchange.js                 统计词频功能，折线图所需
```

views	视图文件
app.js	初始化文件，引入依赖项
package.json	所需库的版本
package-lock.json	所有库的版本与来源

数据库表格：

admin	存放管理员的操作
adminer	存放管理员姓名
eastmoney	存放财经新闻数据
user	存放用户信息
user_action	存放用户操作信息（日志）

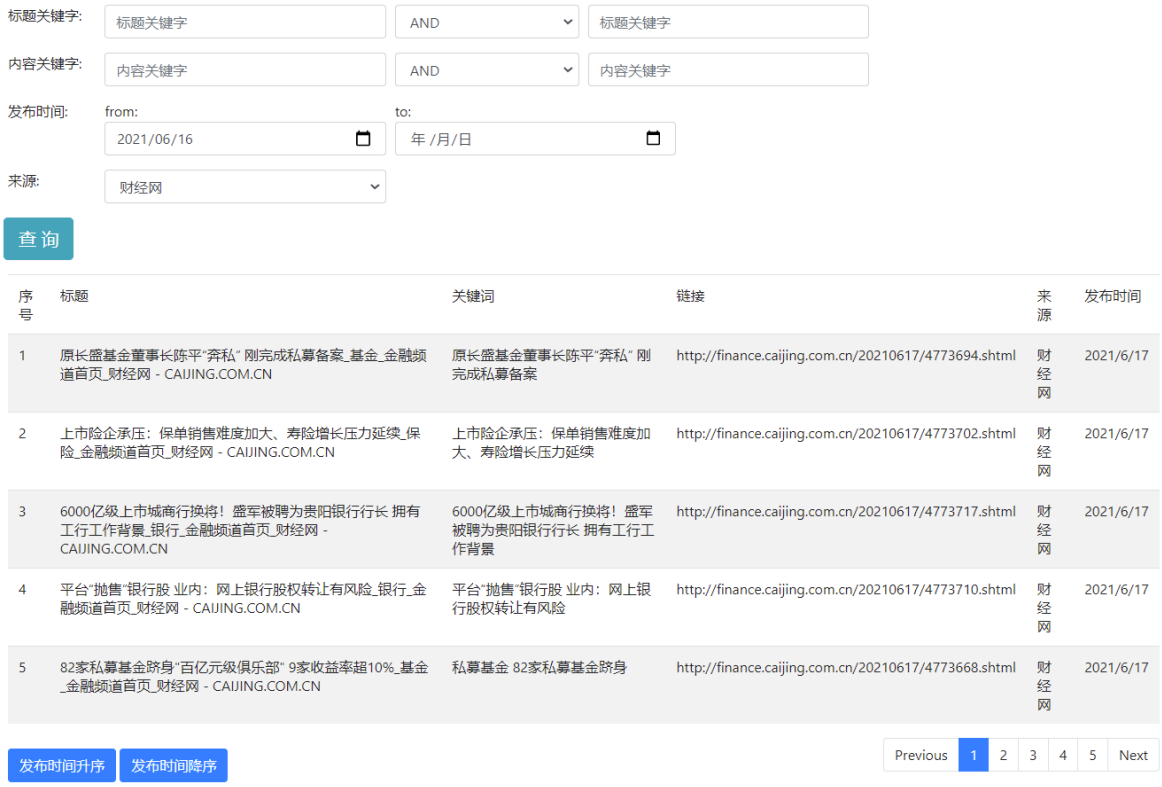
运行方法：在该项目目录下打开cmd，输入node bin/www，之后在网页中输入localhost:3000

（要记得连接自己的数据库，需要更改mysql密码和数据库名称，所需的库需要npm install进行安装，据我的观察使用-g进行全局安装会避免nodejieba安装报错）

功能实现

关于老师已经完成的基本功能我就不再进行详细的阐释了，毕竟老师已经发了视频来展示功能，我就不再截图了。我只展示一下我做了哪些功能上的提升，这些也完全覆盖了项目要求的所有的功能。

一、实现更加丰富的查询功能



从上图可以看到我在"标题关键字"和"内容关键字"的基础上还增加了"发布时间"和"来源"这两个查询功能，这两个功能默认可以都不选，这样就返回全部符合的数据，发布时间可以两个都选，也可以只选择一个，这样更能满足用户的需求。其中"发布时间"这个查询功能在实现的过程中存在一些难点，前端传回的时间格式为：Tue Sep 03 2020 00:00:00 GMT+0800 (中国标准时间)，这时就需要对格式进行处理，否则无法进行sql语句的查询，这需要其他的函数辅助：

```
if( time1 != "null"){
            time1 = new Date(time1);
            time1 = time1.getFullYear() + '-' + (time1.getMonth() + 1) + '-' +
time1.getDate() + ' ' + time1.getHours() + ':' + time1.getMinutes() + ':' +
time1.getSeconds();
        }
```

此外，这里还会遇到一个连锁的问题，由于需要将用户的操作日志保存到数据库，所以这里用户的查询请求也需要放入数据库，但这样的格式很难直接插入数据库中因为它太长了（尤其是当你还选择了其他的搜索功能时），如下所示，数据库中一个单元格最多只能存放50个字符，如果查询请求很长时可能会导致报错。所以也在将用户日志插入数据库之前，需要将字符串进行截取，防止出错。

```
insert into
user_action(username,request_time,request_method,request_url,status,remote_addr)
values('yangshuang', '2021-06-23 22:03:52.189','GET','/news/search?
t1=undefined&ts=AND&t2=undefined&c1=undefined&cs=AND&c2=undefined&time1=Sat%20Ju
n%2012%202021%2000:00:00%20GMT+0800%20(%E4%B8%AD%E5%9B%BD%E6%A0%87%E5%87%86%E6%9
7%B6%E9%97%B4)&time2=Wed%20Jun%2023%202021%2000:00:00%20GMT+0800%20(%E4%B8%AD%E5
%9B%BD%E6%A0%87%E5%87%86%E6%97%B6%E9%97%B4)&source=%E8%B4%A2%E7%BB%8F%E7%BD%91&s
time=undefined','200','::1')
```

```
request_url = request_url.substr(0,30);
```

二、在图表上添加查询功能

如上图所示，折线图展现了关键词在新闻中出现次数的变化情况，我添加了关键词搜索功能，用户可以查看任意词语在时间上的热度，而非仅仅局限于"疫情"这一固定的词汇，而如果用户并不想输入词汇，我默认会直接显示"疫情"这一词汇的折线图。

另外，在实际操作的过程中，我还发现了一些问题，首先在老师给的代码中会使用正则表达式去除新闻中的无用字符，其中使用了"\d\w"去除了单个数字和字母，但实际使用中不适合去除这些内容，比如在去除之后搜索"A股"，则没有内容显示，但是在数据库中又查到了新闻中包含"A股"一词，这样的冲突就会导致一些bug的出现。

其次，我想过使用jieba分词之后再对结果进行显示，但是结合现实的搜索引擎，用户输入的可能并不是标准的词语，而可能是句子等，如果使用分词的方法，可能会导致很多搜索没有结果显示，所以我还是采用了直接匹配的方式，这样虽然慢一些，但是用户搜索的效果会更好。

三、对词云的显示结果进行优化

优化前的词云：

优化后的词云：

上面第一个图是优化词云之前的，使用了正则表达式去除无意义的字符，并使用了jieba分词，然后将结果放入词云，这样做的结果就是未能去除掉停用词，导致词云中最大的词语是"的"、"是"、"在"等常用但无意义的词，这样做出来的词云并不能展现太多的信息，所以我选择对其进行优化。最开始我准备使用加载停用词的方式来去除停用词，但nodejs在加载txt文档的操作上比较复杂，所以我直接用了nodejieba的自带功能提取关键词，我使用了nodejieba.extract语句对新闻的关键词进行抽取，每篇文章取排名前十的词语放入词云中。（事实上nodejieba也有stopword属性，但不知道为什么加载停用词之后几乎没有效果）。最终的结果如第二张图所示，关键词变为了"亿元"、"基金"、"巴菲特"、"公司"等财经词语（我爬取的为财经新闻），可见词云效果比之前要好。

```
var words = nodejieba.extract(newcontent, 10);
```

此外，由于项目要求至少有3个图表，所以在此附上我的第三个图表，直方图：

它展示了新闻发布数随着时间的变化（由于我只在个别的天数爬取了数据，所以这一结果事实上并不能真正显示出新闻发布数的实际情况，但功能是正常的）

四、增加管理员功能

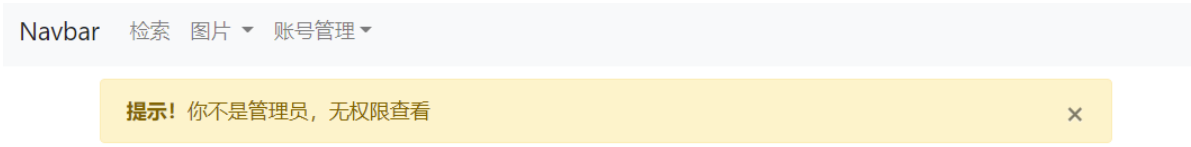
我实现了一个管理端界面，管理员可以查看所有用户姓名、并可以搜索不同用户的操作记录，并能管理（停用启用）注册用户。

这一部分新增的代码比较多，我就不在此实验报告中进行展示了，如有需要可以在github中查看我的项目代码。主要的改动都是在/public/news.html、/public/javascript/news.js、/route/news.js、/dao/regDAO.js这四个文件中进行的，每次对功能的改动都要四个文件一起进行，完成从前端到后端到数据库的完整过程。虽然我之前也有一些做项目的经验，但本次实验还是让我比较痛苦，angular框架虽然让代码变得更有体系但是代码量也变得比较大，并且还要进行一些异常的判断，从而导致我不停的改bug。在不停的改bug的过程中，我也有了一些经验，最好在每个js文件中都不停的使用console.log()来判断代码运行到了哪一步，并且也最好把文件之间传输的数据都打印一下，这其中遇到的bug最多的就是数据类型使用错误、数据没能成功传输、数据接收方式不对等等。

下面主要介绍一下该部分的功能：

1、管理员可以在账号管理部分查看管理员专区，而非管理员则不能查看，如果要查看则会出现警告，如下图所示。

那么是如何设定管理员的呢？如果直接将管理员姓名写在代码中无疑是最简单的，但是如果后续要继续添加管理员那么每次都要修改源代码，这非常的不方便。所以我在数据库中新建了一个管理员的表格adminer，并在mysql中通过insert语句将管理员的姓名插入其中，每当用户想要查看管理员专区，系统就会判断他是否存在于管理员表格之中，如果是则会显示管理员专区，否则不会显示。虽然在mysql中通过insert语句增加管理员稍微复杂了一些，但是如果将增加管理员的功能暴露给前端管理员页面，那么每个管理员都可以无限制的添加别人为管理员，也会出现一些问题，所以我没有将添加管理员功能暴露给前端页面。



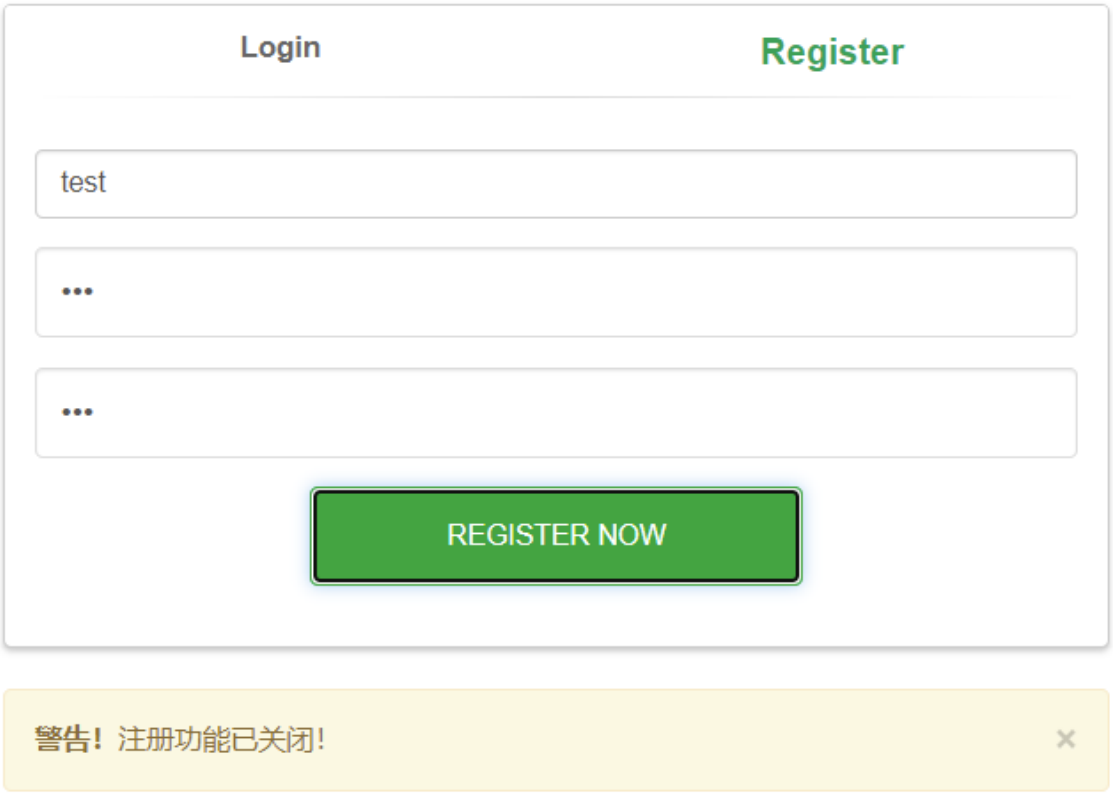
2、管理员可以停用或开启注册功能

为了实现这一功能我也新建了一个管理员操作的表格admin，该表格会记录哪个管理员在什么时候进行了什么操作（这里的操作仅限于停用、开启注册功能），管理员可以不停的按停用或开启注册按钮，但是最终我只会提取表格中管理员最新的操作，sql语句如下所示。当用户进行注册时，我会提取admin表格中最新的操作来判断注册是否开启，如果开启则一切正常进行，如果未开启，则注册功能不能使用，并会给用户一个当前不能注册的提示。

```
SELECT action from admin where actiontime=(select MAX(actiontime) from admin);
```

管理员关闭注册功能：

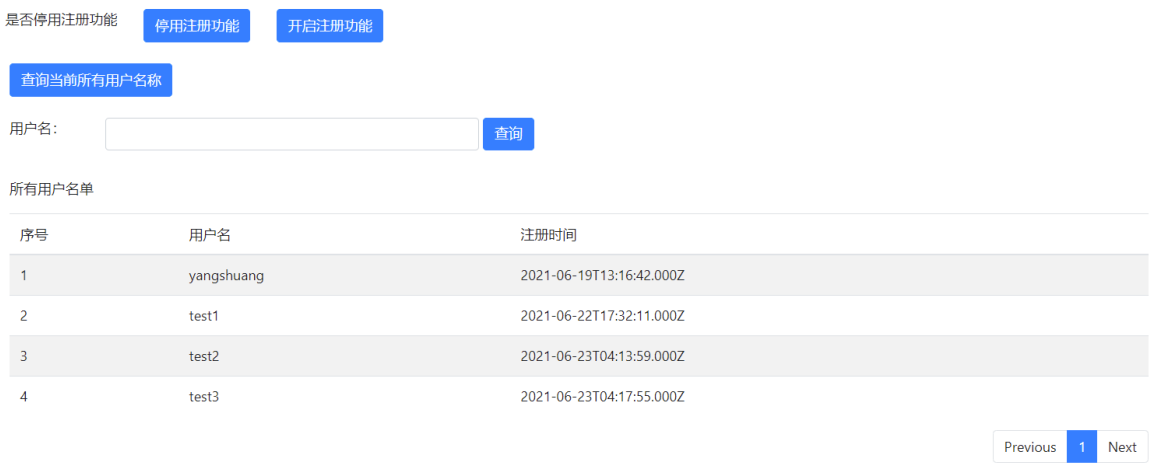
用户无法注册：



3、查看所有用户的姓名信息

当管理员点击"查询当前所有用户名称"按钮时，会显示下图所示的用户信息，这个表格也采用的是分页的方式，只不过我注册的用户比较少，显示不出来分页的样式。这一功能是通过查询数据库中user表格来实现的，我将user表格中所有的用户都放入表格之中，为了保护用户的隐私，这里不会显示给管理员用户的密码。

显示用户姓名的功能主要是为了给后续搜索用户日志信息做铺垫的。如果管理员不知道用户姓名，则无法查看用户日志。

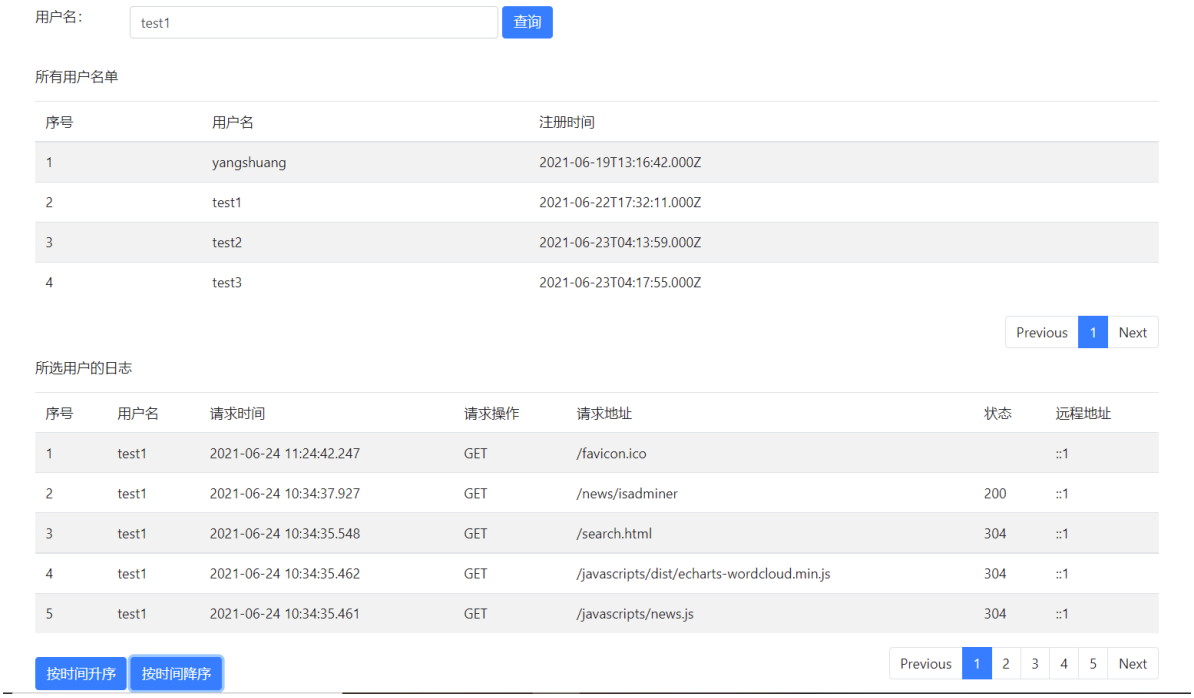


4、搜索用户的日志信息

这一功能使用的是数据库中user_action表格。当管理员在文本框中输入用户姓名并点击查询时，会显示给管理员一张表格（同样也是分页显示），显示用户的日志，管理员可以选择按照用户操作时间进行升序或者降序排列，便于查找。同样的，如果用户不在文本框中输入名字直接点击查询，则会出现所有用户的日志，如果用户输入的用户名称不存在于数据库中，则会返回一个不存在此用户的警告。

这里值得注意的一点是，该表格和用户姓名的表格不能公用一个分页代码，否则会发生奇怪的事情，在html和js文件中一定要将它们的名称全都改的不同。

根据用户名搜索日志：



此外该项目还有一些功能尚未完善，比如bootstrap的alert警告功能在实际使用过程中有一些小问题，当你关闭警告条后，如果你再次执行警告操作时，警告条不会再显示了，只有当你刷新网页再次触发之后才能显示警告条。在我查阅网上资料后发现，如果用户关闭了警告条bootstrap会直接删掉警告条而非隐藏起来，导致只有刷新之后才能再次获得警告条，虽然这不影响基本功能，但可能会使用户体验变差，如果之后有时间我会完善这个问题。

总结

通过该项目我初步接触了angularJS框架，并能照葫芦画瓢完成一些额外的功能，虽然在这个过程中我也遇到了一些困难和不断产生的Bug，但是通过不断的时间与试错，我对数据在前端后端数据库之间的传输有了更深的了解，并能够使用bootstrap库对前端进行一些美化与加工，以及能够查看bootstrap、angular、echarts的官方文档来解决遇到问题。

当然，本项目也有很多值得优化的地方，由于项目完成的比较紧迫，有些代码可能存在冗余，有些接口的具体实现我也还没有完全理清楚，只是照着框架实现了功能，不过好在功能还是比较完善的。