[1. 绪论 2](#_Toc29121)

[1.1互联网技术与教育的联合发展 2](#_Toc4585)

[1.2课题的提出 2](#_Toc10743)

[1.2.1 Node.js 3](#_Toc18860)

[1.2.2 Neo4j数据库 3](#_Toc32417)

[1.2.3 MySQL数据库 3](#_Toc123)

[1.2.4 Ajax技术  4](#_Toc16883)

[1.2.5 Redis数据库 4](#_Toc27878)

[1.2.6 HTML5 5](#_Toc4261)

[1.3论文的组织结构 6](#_Toc8077)

[2.ECloud文件共享平台体系结构 6](#_Toc3224)

[2.1网站功能分析 7](#_Toc8094)

[2.2.网站的系统体系结构 8](#_Toc8650)

[2.2.1 个人面板模块 8](#_Toc19585)

[2.2.2 文件处理模块 9](#_Toc19585)

[2.2.3 登录模块 9](#_Toc20439)

[2.2.4 后台管理模块 9](#_Toc13780)

[2.2.5 游客模块 10](#_Toc1175)

[3.系统子模块技术实现（重点，贴代码） 10](#_Toc25612)

[3.1.网站的构建技术 10](#_Toc14493)

[3.1.1浏览器/服务器模式 10](#_Toc8074)

[3.1.2 数据库的连接 11](#_Toc17560)

[3.1.3网站页面设计 13](#_Toc28893)

[3.2.个人面板模块的实现 15](#_Toc24303)

[3.3文件处理模块的实现 17](#_Toc11398)

[3.4登录模块的实现 17](#_Toc27406)

[3.5后台管理模块的实现 17](#_Toc11648)

[3.6游客模块的实现 18](#_Toc20537)

[3.7 留言模块 18](#_Toc15699)

[4.ECloud的网站实现 21](#_Toc31717)

[4.1 系统环境的配置 21](#_Toc23937)

[4.2系统运行界面 22](#_Toc5091)

[4.3管理员界面展示 33](#_Toc1417)

[5.结论与展望 36](#_Toc16102)

[5.1 总结 36](#_Toc16852)

[5.2 展望 37](#_Toc24102)

[参考文献 38](#_Toc17230)

# **绪论**

## **1.1互联网技术与教育的联合发展**

随着当代互联网产业的不断发展，越来越多的领域应用到了各种高新的互联网技术。而在教育方面也不例外，“互联网+教育”更是这两者所结合产生的一种全新的教育形式。它突破了传统教学的时空限制，使教育变得更加灵活。如北京四中网校平台：将大量高质量的教师教学视频放置到网站平台上，突破了时空限制使学生可以随时随地学习；采用了任务系统与积分兑换制度增强学生的学习积极性；采用了英语单词背诵与pk系统来增强学生的英语能力；开辟了互帮互助区域使学生能够彼此间交流沟通问题等等。

除了这些在中小学生教育平台上的应用外，互联网在各个高校中也扮演着不可忽视的角色。比如为了方便管理学生学籍，方便学生选课与教师排课，了解课程成绩而设计出的高校教育管理系统；为了管理学生的校园卡账户信息而设计的校园卡管理系统；为了管理学校图书而设计的图书管理系统；为了管理课堂学习信息，方便学生上交作业并检查学生出勤情况而设计的课堂管理系统；为了方便了解参与班级课程活动，增强团建建设，帮助大学生寻找合适的志愿工作而大学生活动管理系统等等。这些系统既方便了学校和教师方面的统一安排和管理，也方便了学生的日常学习和生活。由此可见，未来教育与互联网技术的发展将会是密不可分的。

## 1.2课题的提出

随着信息技术的高速发展，人们越来越多地使用计算机进行日常办公，越来越多的文件从纸张变成了存储在计算机中的数据。随着时间的推移，不但数量与规模不断增长，同时也有越来越多的业务需要使用这些文件存储、管理以及随时读取。因此各种文件共享平台,云计算平台应运而生,成为文件管理和分享的主流。

而文档作为传递信息的载体，在互联网和教育方面都起到了重要的作用。许多的大学存在着网络信息资源平台的缺失或缺乏统一管理，没有交互性和统一共享性的问题。通过建立网络文件共享平台，学生和老师可以上传管理和下载所需要的各种文件资源，节省了很多的时间，加快了信息资源交流的速度，有助于提高学生的学习效率。而且通过后台的筛选和管理可以推荐用户可能需要的文件或可能提供这些文件的用户，从而更好地提升资源搜索的效率。

文件共享平台系统作为一种资源的整合手段应该广泛应用到教育行业,为学生提供广泛的学习资料,使学习变得更加便捷,为学生提供自由信息交流的场合。

### **1.2.1 Node.js**

Node.js是一个Javascript运行环境，发布于2009年5月，由Ryan Dahl开发，实质是对Chrome V8引擎进行了封装。Node.js 不是一个 JavaScript [框架](https://baike.baidu.com/item/%E6%A1%86%E6%9E%B6/1212667" \t "https://baike.baidu.com/item/node.js/_blank)，不同于[CakePHP](https://baike.baidu.com/item/CakePHP/5930284" \t "https://baike.baidu.com/item/node.js/_blank)、[Django](https://baike.baidu.com/item/Django/61531" \t "https://baike.baidu.com/item/node.js/_blank)、[Rails](https://baike.baidu.com/item/Rails/10962333" \t "https://baike.baidu.com/item/node.js/_blank)。Node.js 更不是浏览器端的库，不能与 jQuery、ExtJS 相提并论。Node.js 是一个让 JavaScript 运行在[服务端](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E7%AB%AF/6492316" \t "https://baike.baidu.com/item/node.js/_blank)的开发平台，它让 JavaScript 成为与[PHP](https://baike.baidu.com/item/PHP/9337" \t "https://baike.baidu.com/item/node.js/_blank)、[Python](https://baike.baidu.com/item/Python/407313" \t "https://baike.baidu.com/item/node.js/_blank)、[Perl](https://baike.baidu.com/item/Perl/851577" \t "https://baike.baidu.com/item/node.js/_blank)、[Ruby](https://baike.baidu.com/item/Ruby/11419" \t "https://baike.baidu.com/item/node.js/_blank) 等服务端语言平起平坐的[脚本语言](https://baike.baidu.com/item/%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80/1379708" \t "https://baike.baidu.com/item/node.js/_blank)。

### 1.2.2 Neo4j**数据库**

Neo4j是一个高性能的,NOSQL图形数据库，它将结构化数据存储在网络上而不是表中。它是一个嵌入式的、基于磁盘的、具备完全的事务特性的Java持久化引擎，但是它将结构化数据存储在网络(从数学角度叫做图)上而不是表中。Neo4j也可以被看作是一个高性能的图引擎，该引擎具有成熟数据库的所有特性。程序员工作在一个面向对象的、灵活的网络结构下而不是严格、静态的表中——但是他们可以享受到具备完全的事务特性、企业级的数据库的所有好处。

在本文件共享平台中，数据库方面是将neo4j和mysql一起使用，利用neo4j来处理好友推荐和文件资源方面的推荐，而用mysql来处理其他方面的数据管理。Neo4j使用起来更加方便简洁,可视化程度高,适合于关系型数据库。

### 1**.2.3 MySQL数据库**

Mysql是数据库领域的中间派。它缺乏一个全功能数据库的大多数主要特征，但是又有比类似Xbase记录存储引擎更多的特征。它象企业级RDBMS那样需要一个积极的服务者守护程序，但是不能象他们那样消费资源。查询语言允许复杂的连接(join)查询，但是所有的参考完整必须由程序员强制保证。

MySQL在Linux世界里找到一个位置－提供简洁和速度，同时仍然提供足够的功能使程序员高兴。数据库程序员将喜欢其查询功能和广泛的客户库，数据库管理员会觉得系统缺乏主要数据库功能，他们会发觉它对简单数据库(在不能保证购买大牌数据库时)是有价值的。

JDBC2.0提供了javax.sql.DataSource接口，它负责建立与数据库的连接，在应用程序访问数据库时不需要编写连接数据库的代码，可以直接从数据源获得数据库连接。

在DataSource中事先建立了多个数据库连接，这些数据库连接保存在连接池(Connect Pool)中。Java程序访问数据库时，只需要从连接池中取出空闲状态的数据库连接；当程序访问数据库结束，再将数据库连接放回连接池。

DataSource对象是由Tomcat提供的，因此不能在程序中采用创建一个实例的方式来生产DataSource对象，而需要采用Java的另一个技术JNDI，来获得DataSource对象的引用。

Tomcat把DataSource作为一种可以配置的JNDI资源来处理。生成DataSource对象的factory为org.apache.commons.dbcp.BasicDataSourceFactory。

在javax.naming包中提供了Context接口，该接口提供了将对象和名字绑定，以及通过名字检索对象的方法。

**1.2.4 Ajax技术**

Ajax即“Asynchronous JavaScript and XML”（异步JavaScript和XML），它是一种用于创建更好更快以及交互性更强的Web应用程序的技术。JavaScript通过该技术可使用XMLHttpRequest对象来直接与服务器进行通信，可在不重载页面的情况与Web服务器交换数据，AJAX在浏览器与Web服务器之间使用异步数据传输（HTTP请求），这样就可使网页从服务器请求少量的信息，而不是整个页面，可使因特网应用程序更小、更快、更友好。其优点可以归结为：

（1）不必更新全部网页，可更新部分页面。   
 （2）优化了Browser和Server之间的沟通，减少不必要的数据传输、时间及降低网络上数据流量。   
 （3）平衡了前、后端的负载，原本数据大多由后端负责处理，借由AJAX让客户端分担些工作，减低了后端的负载。   
　　本系统对AJAX的使用主要表现是在前台页面上，如用户登录使用AJAX技术验证实现无刷新登录，增加用户的体验。

### 1.2.5 Redis数据库

Redis是一个开源的使用ANSI [C语言](https://baike.baidu.com/item/C%E8%AF%AD%E8%A8%80" \t "https://baike.baidu.com/item/Redis/_blank)编写、支持网络、可基于内存亦可持久化的日志型、Key-Value[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "https://baike.baidu.com/item/Redis/_blank)，并提供多种语言的API。

Redis是一个key-value存储系统。和Memcached类似，它支持存储的value类型相对更多，包括string(字符串)、list(链表)、set(集合)、zset(sorted set --有序集合)和hash（哈希类型）。这些数据类型都支持push/pop、add/remove及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，Redis支持各种不同方式的排序。与memcached一样，为了保证效率，数据都是缓存在内存中。区别的是Redis会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了master-slave(主从)同步。

Redis 是一个高性能的key-value数据库。 Redis的出现，很大程度补偿了memcached这类key/value存储的不足，在部 分场合可以对关系数据库起到很好的补充作用。它提供了Java，C/C++，C#，PHP，JavaScript，Perl，Object-C，Python，Ruby，Erlang等客户端，使用很方便。

### **1.2.6 HTML5**

HTML5 是一种 WEB 标记语言， 主要用于开发网页使用。HTML5 是 WEB 应用中一种“超文本标记语言（HTML）”的第五次重大修改， 我们将这次修改后的 HTML 标准 ， 称之为“HTML5”。

HTML5 就是包括 HTML、CSS 和 JavaScript 在内的一套技术组合，进一步加强了页面的表现性能。 所以说未来 HTML5 将成为 HTML、XHTML 以及 HTML DOM 的新标准。

HTML5的优势在于：

（1）解决了跨浏览器问题

在 HTML5 之前，各大浏览器厂商为了争夺市场占有率，会在各自的浏览器中增加各种各样的功能，并且不具有统一的标准。 使用不同的浏览器，常常看到不同的页面效果。 在 HTML5中，纳入了所有合理的扩展功能，具备很好的跨平台性能。

（2）更低的开发成本，更精美的动画效果

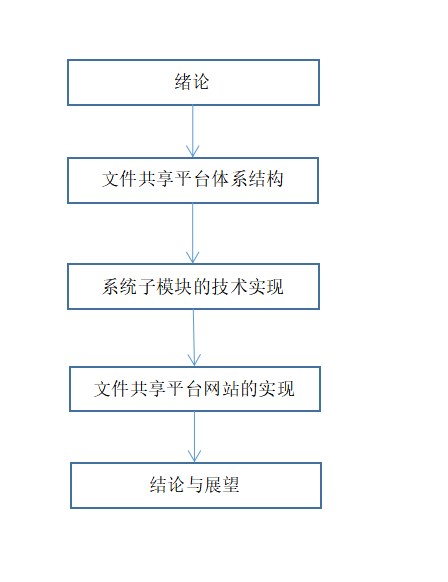
使用 HTML5 标签和 CSS 层叠样式所实现的动画效果应用到网页中会使人眼前一亮， 并且 HTML5 技术能够实现开发一次，可同时应用到手机、网页平台，实现“一次设计，普遍适用，从而降低企业开发成本。

（3）化繁为简的优势

新的简化的字符集声明、新的简化的 DOCTYPE、简单而强大的 HTML5 API， 以浏览器原生能力替代复杂的 JavaScript 代码。为了实现这些简化操作，HTML5 规范需要比以前更加细致、精确。 为了避免造成误解，HTML5 对每一个细节都有着非常明确的规范说明，不允许有任何的歧义和模糊出现。

## **1.3论文的组织结构**

论文在第一章介绍了互联网与教育的联合发展，介绍了ECloud文件共享平台建设的背景，介绍了网站平台实现的基本技术；在第二章中描述了网站的整体结构，介绍了网站内容，以及各部分实现模块的划分。在第三章中详细介绍了网站各个模块实现所用到的技术，介绍了如何运用这些技术来实现各模块的功能。在后面的第四章中，介绍了ECloud文件共享平台的界面展示，并具体介绍了操作方法和实现的功能。而在最后的第五章中，我们总结了ECloud文件共享系统的开发过程并分析系统还有那些不足之处或提升空间。



# 2.ECloud文件共享平台体系结构

## 2.1网站功能分析

ECloud文件共享平台的角色总共分为三个：未登陆的游客，已登陆的用户，已登陆的管理员

ECloud对游客开放的功能有：

\* 用户注册 在ECloud平台上创建一个新用户并使用ECloud系统的各种功能

\* 忘记密码找回 通过注册时填写的邮箱来修改自己的密码

ECloud对用户开放的功能有：

\* 文件上传 上传文件和各种资料到平台上进行分享

\* 文件下载 从平台上进行下载

\* 分享推荐 根据所关注的人和下载的文件来推荐用户可能需要的文件以及可能感兴趣的人

\* 资料修改 修改他人所见的个人资料，修改密码等等

\* 上传管理 对上传文件的帖子进行修改和删除等操作

\* 操作记录 可以对之前所进行的部分操作或造成积分增减的原因进行查询

\* 留言功能 在上传文件的下方留下文字，方便其他下载的人对下载的内容了解。也可以与他人进行对话方便信息交流，可以对留言进行点赞。在自己上传的文件下也可以进行对留言的管理

\* 我的留言 对自己发表的评论或回复可进行删除操作

\* 关注他人 对上传的人或者好友进行关注,方便得到有关更多他们的信息

\* 个人面板 对个人资料进行修改 管理上传文件 关注的人等等的功能页面

\* 他人面板 可以看到他人的部分个人资料，并看到对方上传的文件等等。

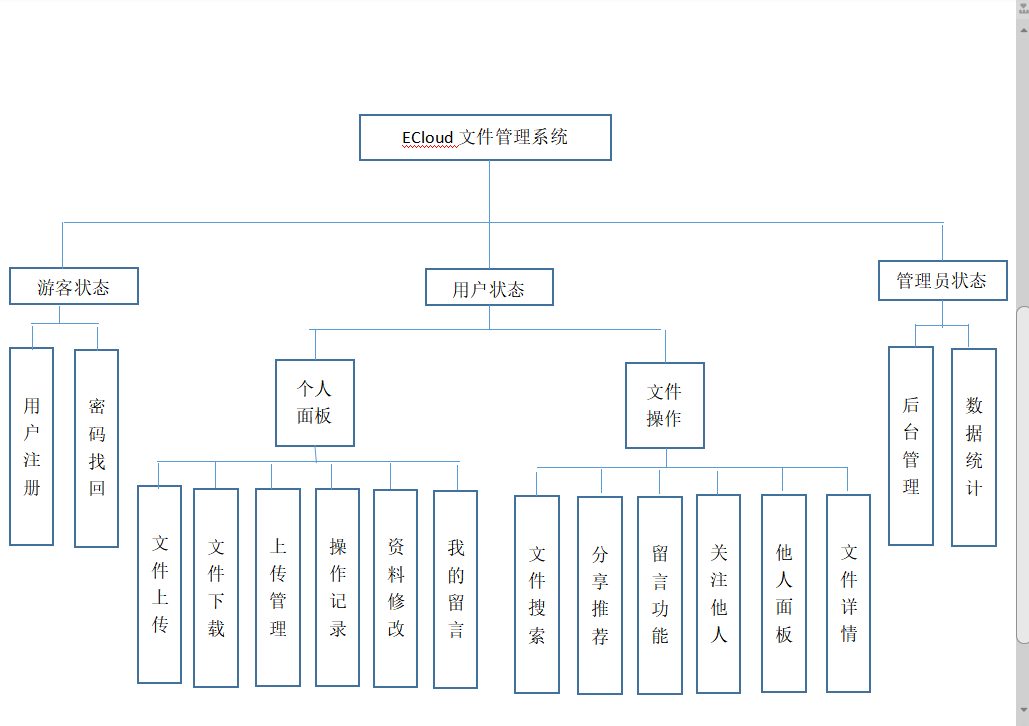
\* 资源检索 从平台上检索所感兴趣的文件

\* 文件详情 可以看到目标文件的标题，大小，下载所用积分，上传者以及他人对文件的评价等

ECloud对管理员开放的功能有：

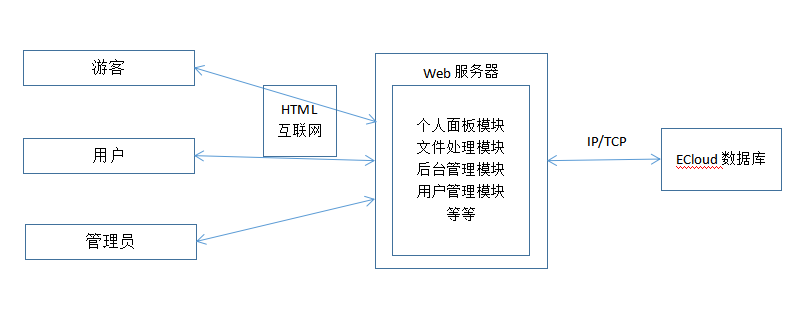
\* 数据统计 对网站平台的用户类型分布和文件类型分布进行统计

\* 后台管理 对文件和用户进行管理



## 2.2.网站的系统体系结构

网站通过当前用户角色和角色想要执行的功能或者说所停留的页面进行了大致的分类。具体可以分为三个角色。角色中的用户还可以继续细分更多。包括个人面板模块，文件处理模块等。而管理员则有用户管理模块。

**2.2.1 个人面板模块**

本模块主要用于处理用户的个人信息，与文件的处理模块相比更着重与账户本身所具有的信息查询以及处理。

具体功能包括个人信息修改，文件上传，文件管理，操作记录查询，积分规则说明，关注的人列表查看，密码修改，我的留言及退出登录几项。

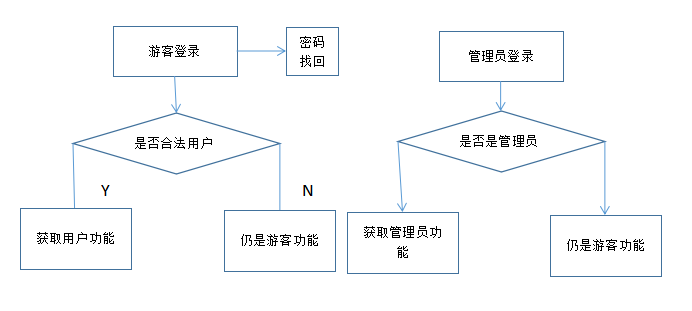
### 2.2.2 文件处理模块

文件处理模块是基于用户对文件的各种方面的需求所设计的。其中包括用户对文件的直接搜索需求，用户对文件的简介推荐需求，用户对文件具体详细信息的获取，以及对相关文件进行评价以使其他寻找类似文件的人节省一定搜索下载的时间。



### 2.2.3 登录模块

本模块主要通过数据库内所存储的账号信息验证用户的身份，将系统当前状态由游客状态转换为用户状态或是管理员状态，从而使使用者体验到本系统的更多功能。相关流程如下图所示：



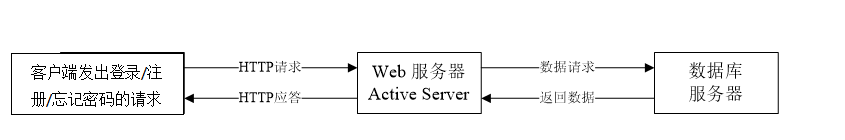
### **2.2.4 后台管理模块**

本模块主要用于管理员对后台数据的管理，具体包括网站数据的统计以及对网站用户和资源数据的管理。模块的运行流程如下所示。



### **2.2.5 游客模块**

本模块是为了给未使用过ECloud的用户用户身份，或给拥有ECloud用户但却忘记ECloud账号密码的人设立的。在本模块中，游客可以直接登录自己的账号或者注册一个账号。而忘记密码的游客可通过注册时填写的邮箱信息来修改账号的密码使自己可以重新登录自己的账号。具体的流程如下所示



# 3.系统子模块技术实现（重点，贴代码）

3.1.网站的构建技术

### 3.1.1浏览器/服务器模式

B/S模式也就是通常所说的浏览器／服务器模式，B/S模式是把在前端采用ＩＥ浏览器将用户提交的操作信息向WEB服务器发出HTTP 请求，WEB服务器通过ASP和一些中间组件访问后台数据库，并将操作结果以HTML页面的形式返回给前端浏览器。目前，B模式逐渐成为了开发各种应用系统的最常用的一种方式。B/S模式与传统C/S（客户机／服务器）都是以同一种请求和应答方式来执行应用的。但传统的C/S是一种二层或三层结构模式，其客户端需要集中大量的应用软件，而 B/S客户端仅需单一的浏览器软件，是一种全新的体系结构。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护 和使用。降低了使用成本，平台的选择面更广，服务器不仅可以是广泛的windows家族产品，也可以选用网络安全性较好且几乎为零的Linux平台。而客户机上只要安装一个浏览器就能实现软件应用。由于有了这些显而易见的优点，越来越多的考试，越来越多的比赛、越来越多的教学开始基于B/S模式搭建，如会计职称考试，会计从业考试，各种市场营销和物流的教学及比赛软件，尤其是一些不需要大量调用本地资源的软件，几乎都是基于B/S模式搭建的[]。

浏览器的客户机/服务器(Browser/ Server,B/S) 结构通信方式比C/ S结构高效。B/S结构将系统功能实现的主要内容集中到服务器上。客户端可以使用任何一个浏览器如 Internet Explorer 或 Netscape Navigator,数据库服务器安装 H2、SQL Server、Oracle、MYSQL 等数据库。浏览器图1B/S模式结构图可以通过 Web 服务同数据库进行数据交互B/S模式结构图如图 1所示。



采用B/S模式有以下的好处:

(1)各个用户通过HTTP请求在权限范围内调用Web服务器上不同处理程序,从而完成对数据的查询或修改;

(2)它使用户的操作变得更简单;

(3)客户端只是一个简单易用的浏览器软件。无论是决策层还是操作层的人员都无需培训,可以直接使用。B/ S模式的这种特性,还使系统维护的限制因素更少; B/ S 也适用于网上信息发布,使得传统的MIS的功能有所扩展[2]。

### 3.1.2 数据库的连接

Neo4j数据库：

var db = new neo4j('http://neo4j:\*\*\*\*\*\*@localhost:7474');

Var cypher = ‘match (n) where n.id=796 return n’

db.cypherQuery(cypher, function(err, result) {

if(err) {

throw err;

return false;

}

res.end(JSON.stringify(result.data));

});

Mysql数据库：

mysql = require('mysql'),

pool = mysql.createPool({

'host': 'localhost',

'port': '3306',

'user': 'root',

'password': '\*\*\*\*\*\*',

'database': 'eloud',

});

sql = ‘select \* from comment where id=89’

pool.getConnection(function(err, connection) {

if(err) {

throw err;

return false;

} else {

connection.query(sql, function(err, result) {

if(err) {

throw err;

return false;

} else {

connection.release();

res.end(JSON.stringify(result));

}

})

}

});

Reids部分（主要用于找回密码，不作重点）：

redis = require('redis'),

var rds\_port = 6379,

rds\_host = '127.0.0.1',

rds\_opts = {};

client = redis.createClient(rds\_port, rds\_host, rds\_opts);

client.on('error', function(err) {

if(err) {

throw err;

return false;

}

});

client.on('ready', function(err) {

if(err) {

throw err;

return false;

}

});

client.on('connect', function(err) {

if(err) {

throw err;

return false;

}

client.exists(token, function(err, response) {

if(response == 1) {

var cypher = `match(user:User)

where user.username=${username}

set user.password=${new\_password}`;

db.cypherQuery(cypher, function(err, result) {

res.end(JSON.stringify({state:'success'}));

});

} else {

res.end(JSON.stringify({state: 'fail'}));

}

});

})

### 3.1.3网站页面设计

一个好的网站除了满足用户需求之外，界面友好也是相当重要的。在进行系统设计之前，需要列出功能需求，并将功能需求进行细致的分类，按照大众的使用习惯，将这些功能进行精心的布置，分块，使用户拥有良好的使用体验。而网站页面设计的具体操作则是通常根据用户的接收能力先进行信息分块，其次根据信息结构安排分割页面，然后编织页面间的链接，最后设计风格统一的版面。

（1）信息分块

信息分块，在站点建设过程中是一件非常有用的事情。通过信息分块，我们能够清晰地把握整个站点的结构，使我们的网站能够以更加精妙的结构来覆盖更多的信息。在设计Web页面时我们必须对信息分块进行认真地考虑，因为人的大脑在一定时间内接收的信息量是有限的。根据研究表明，在一个Web页面中，用户一次接收的信息量以五段左右为宜。过量的信息容易导致人脑的注意力分散，影响正常的理解能力和记忆能力。因此，在设计Web信息系统时进行信息分块，主要是为了使Web页面提供的信息量在便于用户理解和记忆的限度以内；有利于建立可以重复使用的Web信息页面，提高页面利用率；有利于用户对Web页面描述的内容集中注意力，以渐进的方式积累信息。

在ECloud系统中，首先根据角色的分配的不同，给不同的角色以不同程度的信息。其次是根据操作对象的不同将页面分为了两大类，一类是处理账号相关的页面，而另一类则是处理文件相关的页面。经过这样的信息分块，用户就可以大致了解怎样操作可以接近自己想要的页面。

（2）版面设计与布局

版面设计，可以看作是网页“空间的调度”。在Web的界面设计中，Web页面实际上就是文字和图像所传达的信息的活动空间。而怎样在屏幕这一方寸之地，借助空间的经营序化内容，有效地传达信息，也并不如想象的那么简单，因为“屏幕”从本质上来讲并不适合阅读，怎样把这种客观上的不适转变成主观上的视觉愉悦呢，这就依赖于对各种视觉技法的应用了。

Web页是一个具有顶端和底部、包含内容的实体，它没有大小的限制，这与实际的纸张页面有所区别。一个Web页中可能只有几个字，也可能有一百万个字。

在Web页上所放内容的多少将会影响该Web页被阅读的方式。在Web中用指针链接的其它页能极大地扩展单个Web所能包含的信息。在一个Web页上内容的大小是没有限制的，所以把所有的内容都放到一个Web页上也并非不可能，这只是意味着浏览者必须把它作为一单页而从头读到尾。如果将所有的信息分成多个部分用指针分别指向它们，那么不愿看的部分就可以略过。在一个Web页中内容放多放少主要取决于设计目的及所需效果，而不是Web页的物理大小或字体大小。Web页大小的灵活性使web页上所放内容多少应按每一具体实例而定。页的长度或大小依据页的内容和目的而变化。

放置信息的方法，用大的还是用小的页面，依赖于Web站点的内容和结构。仅有的限制是当前技术的标准——计算机监视器的大小，以及通过Internet传输大量信息和处理非常大的页面所需要的花费。页面大小应该与需要的一样长或短。有可能不跳转到新页的顶部，而是跳到页上指定的某一点。这个内部链接可以使一米长的文本流很有用。然而，设计得差的一米长的文本流，将会很难读，消化时会碰到困难，导航时会遇到问题。纸张有一个预先确定的顶部和底部——如果浏览者在一个非常长并且设计得差的Web页上翻滚时，他会变得迷惑并且迷失方向。如果完全不限制大小，那么好的信息设计和清楚的导航帮助变得非常重要。根据实际情况，和大多数人的喜好，页面大致都在一屏到一屏半的长度。

虽然由于技术上的原因，网页上的文字与图片不能自由的叠加组合，但仍然可以借鉴平面设计的方法，把版式设计的不拘一格。总的来说，网页的版式可分为规则型和不规则型。

而总的来说，ECloud系统中的网页板式趋向于规则型，一般以单页一屏幕的信息为主，例外的页面信息一般也不会超出两屏幕，避免了用户的过多翻页操作。而且各种信息查询结果的页面也都配备的翻页和跳页功能，方便用户的正常使用。而布局部分，网页的个人面板模块中将按钮大部分都放置与偏右的位置，因为大部分人都是右撇子，将按钮放在右边更符合大众的操作习惯，可能会感觉操作时更加舒适一些。

（3）网页配色

网页配色也是web网页设计的一大重要组成部分，因为人通过视觉来观察网页，而人的视觉又对颜色十分敏感。除此之外，色彩甚至还会对人情绪产生影响，这些内容诸多研究中都有所描述，这里将不再赘述。

色彩在网页中的配置和使用，首先应该使各个颜色间保持协调，让人感到舒适，可以长时间地沉浸于网页之中。就比如说如果网页使用大片的红色，并使人长时间盯着这个页面的话，会使人的眼睛更容易产生疲劳。这不是正常网页设计想要达到的效果。而其次应稍微添加些重点，人在浏览东西时总会先寻找一个东西作为焦点，而在网页中，我们可以通过使用小部分使用鲜亮的颜色来人为地制造这种焦点，同时也可以将用户的注意力集中到这部分的信息上。

本网站主要使用了深灰，浅灰和淡蓝，白色这些纯色进行搭配。利用各个颜色的亮度不同来形成不同的层次，web页面中对页面进行了功能块的分割。这些颜色彼此间形成对比，分割开来的页面也显得简洁分明。而部分元素也使用了圆角的设计，圆和方的相互碰撞使网站的总体风格显得不那么生硬。而网站的logo上也跳跃地使用了鲜绿色，可以起到吸引人注意力的作用。

所有的网页都按照以上的方向进行设计，使用同一的配色风格，配图风格和布局风格，有利于用户保持清晰的操作思路。

## 3.2.个人面板模块的实现

本模块使用了Neo4j数据库和Mysql数据库。Neo4j数据库主要用于管理用户和管理员类数据和资源类数据,用Neo4j数据库可以方便了解各个用户对于文件的爱好和需求,从而筛选出符合用户爱好或需求的文件甚至相似的用户来进行推荐,方便用户进行资源查找。Mysql数据库主要用来管理、记录用户的积分变化信息，包括上传资源、下载资源、登录操作、注册操作在内的各种操作均能引起用户积分的变化。相比于用户之间的互相关注（Follow），用户上传下载资源、登录等操作频率较高。考虑到Neo4j数据库中，单个节点与外部节点产生较多关系时，即单个节点连接较多边时，对该单个节点的插入和查询速度将会降低很多。因此，该系统中我们采用Neo4j数据库和Mysql数据混用的方式，在保证提供推荐用户和资源功能的前提下，尽可能提高数据库的查询和更新速度，改善用户体验。

Neo4j数据库中，有十三类标签，个人面板部分涉及到十个标签分别为：

**表3.2.1 标签:User及其属性**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 中文描述 | 类型 | 是否可为空 | 是否主键 |
| id | 数据库系统分配唯一id | 数值型 |  | √ |
| username | 用户名 | 字符串 |  | √ |
| telephone | 手机号 | 字符串 |  |  |
| sex | 性别 | 字符串 |  |  |
| point | 积分 | 数值型 |  |  |
| password | 密码 | 字符串 |  |  |
| motto | 个性签名 | 字符串 | √ |  |
| lastLoginTime | 上次登录时间 | 字符串 |  |  |
| email | 邮箱 | 字符串 |  |  |
| campusCardId | 学生证号 | 字符串 |  |  |
| IDCard | 身份证号 | 字符串 |  |  |

**表3.2.2 标签:Source:Courseware等9个标签及其属性**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 中文描述 | 类型 | 是否可为空 | 是否主键 |
| id | 数据库系统分配唯一id | 数值型 |  | √ |
| description | 文件描述 | 字符串 |  |  |
| downloadCount | 下载次数 | 数值型 |  |  |
| filename | 文件名 | 字符串 |  |  |
| path | 文件路径 | 字符串 |  |  |
| point | 下载该文件所需积分 | 数值型 |  |  |
| size | 文件大小 | 数值型 |  |  |
| title | 帖子标题 | 字符串 |  |  |
| viewCount | 浏览量 | 数值型 |  |  |

Mysql数据库中，表opr 用于记录用户的操作时间,操作内容,消耗或获得的积分以及具体的上传或下载的内容等。id和time一起作为该表的主键。

**3.2.3表opr的字段**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文描述 | 类型 | 长度 | 是否可为空 | 是否主键 |
| id | 用户名 | varchar | 20 |  | √ |
| time | 操作的时间 | datetime | 0 |  | √ |
| opno | 操作类型号 | tinyint | 4 |  |  |
| info | 操作信息 | varchar | 255 | √ |  |
| point | 积分 | tinyint | 4 |  |  |

Mysql数据库中，表comment 用于记录用户的操作时间,操作内容,消耗或获得的积分以及具体的上传或下载的内容等。id和time一起作为该表的主键。

前端方面，为了用户的体验而选择了便签的方式，将各种功能安置到不同的标签上。个人面板的具体功能如下：

1. 个人资料查看修改功能：通过用户名调取用户的相关信息，从而查看到个人资料。修改则是在前端按下修改按钮后，在对应的位置输入想要修改的信息，按下提交后，网页将修改后的信息发送给服务器，服务器则完成对数据库中相应信息的修改。
2. 文件上传功能：在网页中输入想要上传文件的相关信息，并将文件和这些信息一起上传的服务器上去。服务器对文件的大小信息进行获取并和其他填写的信息一起保存到数据库中，并在opr表中生成一个上传文件的记录，对用户的积分进行修改更新。
3. 文件管理功能：通过用户名在资源表中查询由当前用户所上传的文件信息并显示到前端上，前端可以对这些文件进行浏览和查看。如果想删除文件则是按下前端的删除按钮，网页就会的服务器发出一个删除资源表中对应文件的请求，服务器收到请求后会执行对数据库资源表对应的元组的删除工作，并在opr表中生成一个删除文件的记录，对用户的积分进行修改更新。
4. 积分记录功能：在数据库opr表中搜索用户名为当前用户的记录，并显示到前端上，方便用户对账号的相关操作信息进行查询。
5. 积分说明功能：在前端介绍积分的使用获取方法等，方便用户的使用。
6. 关注的人功能：查看自己已经关注的用户，同时可对自己关注的用户进行取消关注操作。
7. 修改密码功能：为了保护用户账号不会被其他人恶意修改，采用的是输入旧密码和新密码的方式进行修改。同样先使用前端对密码格式进行检查，通过检查后发送请求检查旧密码和数据库中密码是否匹配，匹配则进行更新密码的操作。
8. 我的留言功能：可查看自己的所有留言，并可对自己的留言进行查看详情（查看他人对此条留言的回复）、回复、删除、点赞等操作。
9. 退出登录功能：退出登录状态，页面返回到登录页面。

## 3.3文件处理模块的实现

1. 文件搜索功能：关键词搜索，正则查询,并通过levenshtein() 函数返回正则匹配结果与搜索结果之间的 Levenshtein 距离，以此对搜索结果进行排序。
2. 文件/好友推荐功能：关系数据库neo4j通过用户的下载记录和交互推算出用户的爱好偏向，并将相关的文件或是其他用户推荐出来。
3. 文件详情查看功能：通过资源表source导出文件的相关信息。
4. 文件下载功能：网页向服务器发送请求，将服务器上上传好的文件下载到客户端。
5. 留言交互功能：使用了一张留言表，运用树形结构，完成用户间的交流功能。该功能的重点在于如何实现直接在文件下新建一个留言和实现对留言进行回复。同时，该部分还可以完成点赞和删除留言的操作。

## 3.4登录模块的实现

（1）用户登录功能：用户登录分成两部进行，第一步是前端的检查，检查用户是否有填写用户名，密码和验证码，输入的验证码和生成的验证码是否匹配。通过前端的检查后再将输入的用户名和密码信息发送给服务器，服务器检查用户名和密码是否匹配，匹配则登录成功，不匹配则登录失败并显示登录失败信息。

（2）管理员登录功能：与用户登录功能类似

## 3.5后台管理模块的实现

Neo4j数据库中建有Admin管理员标签用来存储管理员账号的相关信息,方便对系统的管理与维护。

**表3.5.1 标签:Admin及其属性**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 中文描述 | 类型 | 是否可为空 | 是否主键 |
| id | 数据库系统分配唯一id | 数值型 |  | √ |
| adminame | 管理员名 | 字符串 |  | √ |
| password | 密码 | 字符串 |  |  |
| authority | 权限 | 字符串列表 |  |  |

Mysql数据库建立两张表,表名分别为census和minifo,用于统计ECloud的用户和文件上传下载量以及用户地区分布。表census 用于分析用户所在的省市并绘制出用户在全国内的分布图。其中censusno为用户身份证号的前两位。

**表3.5.2 表census的字段**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文描述 | 类型 | 长度 | 是否可为空 | 是否主键 |
| censusno | 省市的编号 | Varchar | 4 |  | √ |
| censusname | 省市的名称 | Varchar | 20 |  |  |
| longitude | 省市的经度 | Varchar | 20 |  |  |
| latitude | 省市的纬度 | Varchar | 20 |  |  |

表minifo 用于将id和省市号连接起来，minifo表中的censusno为census表的外码。

**表3.5.3 表minifo的字段**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文描述 | 类型 | 长度 | 是否可为空 | 是否主键 |
| id | 用户名 | Varchar | 20 |  | √ |
| censusno | 省市的编号 | Varchar | 4 |  |  |

1. 数据统计功能：查看ECloud文件管理平台的相关用户和资源的信息统计，将数据库中的信息进行整合后利用百度的echarts将数据整合成图表显示在对应网页上。
2. 后台数据管理，极端情况下，管理员可直接删除数据库中的用户和资源。

## 3.6游客模块的实现

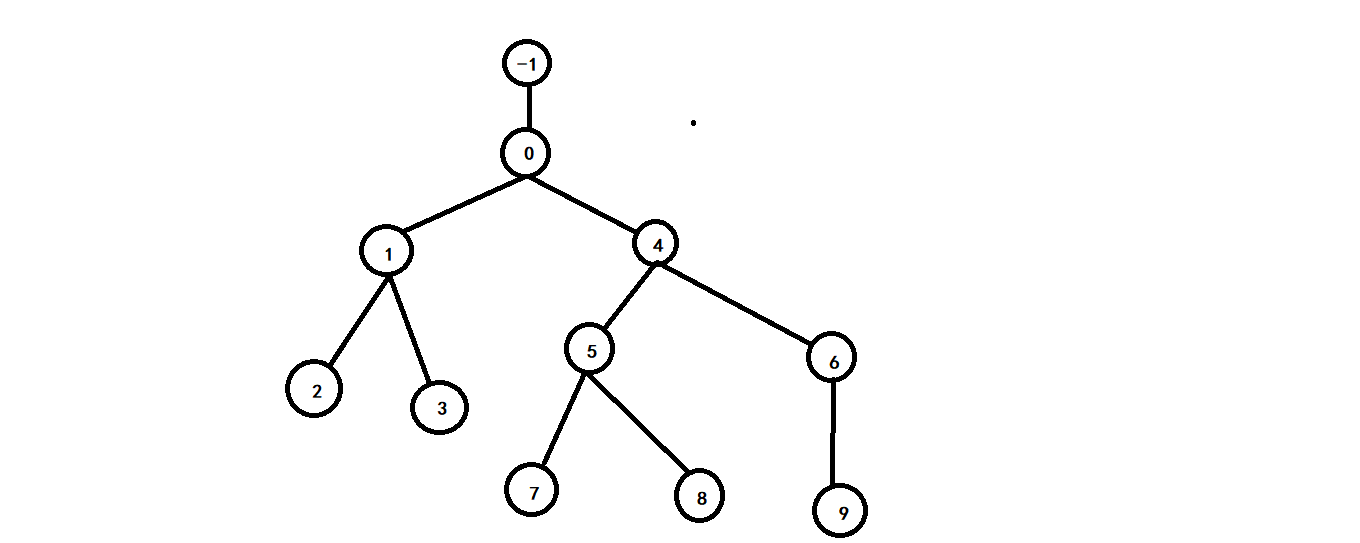
（1）用户注册功能：用户注册功能需要用户先在设计好的前端中填写自己的账号和个人信息，再由前端检查输入的信息是否符合规范。网页将这些账号信息发送给数据库从而实现创建一个新的账号。

（2）密码找回功能：用户在进行注册时有填写邮箱信息，用户可以通过在密码找回界面填写自己的邮箱的地址。网页将邮箱地址提交给数据库，数据库在用户中找到匹配的邮箱后，向对应的邮箱发送一封可以修改该账户密码的网址。在这个网址中输入新的密码，账户的密码就会被修改，用户就可以使用修改过的密码登录自己的账号了。

## 3.7 留言模块

Mysql数据库中存有两张表，comment和user\_like，前者为评论表，后者为用户点赞表，一个由comment和user\_like组合的视图-like\_num，一个过程-insert\_and\_return\_id，以及一个函数getChildren。

考虑到，一条留言或回复只能对至多一条留言或回复生效，当用户直接对帖子直接进行留言时，该留言的父节点默认为-1（虚节点），当用户对特定留言或回复进行回复时，用户的回复的父节点即为所回复的节点。因此，对于留言或回复我们采取树状结构，如图：



**图3.7.1 留言回复树状图**

**注**：节点中的数字即为评论或回复id

**表3.7.1 表comment的字段**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文描述 | 类型 | 长度 | 是否可为空 | 是否主键 |
| id | 评论id | int | 20 |  | √ |
| pone | 留言人用户名 | varchar | 20 |  |  |
| time | 留言时间 | datetime | 6 |  |  |
| src | 资源id | int | 8 |  |  |
| text | 留言内容 | varchar | 255 | √ |  |
| parentnode | 留言的父节点 | int | 20 |  |  |

**注：**

字段src作为索引，索引类型为NORMAL，索引方法为BTREE。

**表3.7.2 表user\_like的字段**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文描述 | 类型 | 长度 | 是否可为空 | 是否主键 |
| username | 用户名 | varchar | 20 |  | √ |
| comment\_id | 评论id | int | 20 |  | √ |

**注：**

字段comment\_id作为索引，索引类型为NORMAL，索引方法为BTREE。

字段comment\_id作为外键，参考comment表中的comment\_id字段。

**视图：like\_num**

select `comment`.`id` AS `id`,`comment`.`pone` AS `username`,`comment`.`time` AS `time`,`comment`.`src` AS `src`,`comment`.`text` AS `text`,count(`user\_like`.`username`) AS `like\_num`,`comment`.`parentnode` AS `parentnode` from (`comment` left join `user\_like` on((`user\_like`.`comment\_id` = `comment`.`id`))) group by `comment`.`id` order by `like\_num` desc,`comment`.`time` desc

**表3.7.3 视图like\_num的字段**

|  |  |
| --- | --- |
| 字段名 | 中文描述 |
| id | 评论id |
| username | 用户名 |
| time | 留言时间 |
| src | 资源id |
| text | 留言内容 |
| Like\_num | 点赞数 |
| parentnode | 留言的父节点 |

**过程：insert\_and\_return\_id**

CREATE DEFINER=`root`@`localhost` PROCEDURE `insert\_and\_return\_id`(in pone varchar(20),in time datetime(6), in src int(8), in text VARCHAR(255), in parentnode int(20),out oid int)

BEGIN

insert into comment (pone, time, src, text, parentnode) values (pone, time, src, text, parentnode);

select max(id) from comment into oid;

select oid;

END

该过程的功能为，当用户发表留言或者回复他人时，用户的留言或回复插入数据库comment时，数据库系统立即返回一个当前插入节点的id，通过后端发送给前台，做到用户留言之后即可看到自己的留言，并可对留言进行点赞等操作的效果，提高用户体验。同时，通过控制前台页面滚轮，评论之后立即回滚到当前留言，进一步提升用户体验。

**函数：getChildren**

CREATE DEFINER=`root`@`localhost` FUNCTION `getChildren`(comment\_id INT) RETURNS varchar(4000) CHARSET utf8

BEGIN

DECLARE oTemp VARCHAR(4000);

DECLARE oTempChild VARCHAR(4000);

SET oTemp = '';

SET oTempChild = CAST(comment\_id AS CHAR);

WHILE oTempChild IS NOT NULL

DO

SET oTemp = CONCAT(oTemp,',',oTempChild);

SELECT GROUP\_CONCAT(id) INTO oTempChild FROM `comment` WHERE FIND\_IN\_SET(parentnode,oTempChild) > 0;

END WHILE;

RETURN oTemp;

END

该函数的只要功能为，当用户删除其留言或回复时，前台传给后端一个用户留言或回复的id，数据库系统通过该函数找到所有以该条留言或回复为父节点的子留言、子回复id（包含该条留言、回复id），进一步删除包含该条评论在内的所有节点。

如图3.7.1留言回复树状图，如删除id值为4的节点，getChildren函数返回的id为：4、5、6、7、8、9，进一步可以删除如上节点。

留言模块的整体功能如下：

（1）用户可在他人文件下进行留言评论，同时也可以对他人留言进行回复。

（2）用户可对他人留言进行点赞操作，同一用户只能对一条留言或回复进行单次点赞操作。

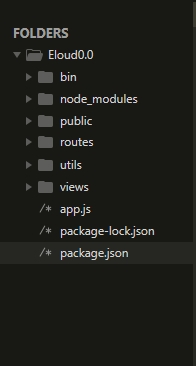
（3）用户可点击相应提示Button查看留言或回复下的多级回复。

（4）用户可对自己的留言进行删除操作。

# 4.ECloud的网站实现

## 4.1 系统环境的配置

系统所需的配置及工具：整个系统采用Express框架，编辑工具为Sublime Text。整个系统的架构如下：



**图4-1 配置图1**

其中：

bin文件夹：存放服务器的配置信息。

node\_modules：存放后端使用的各种包模块。

public文件夹：存放包含html,javascript,css,pictures等在内的前端页面元素。

routes文件夹：存放后端路由。

utils文件夹：存放后端使用的工具包（自定制）。

## 4.2系统运行界面

① 登录页面



头部的轮播banner用来推送和广告。

下面的部分填上logo。

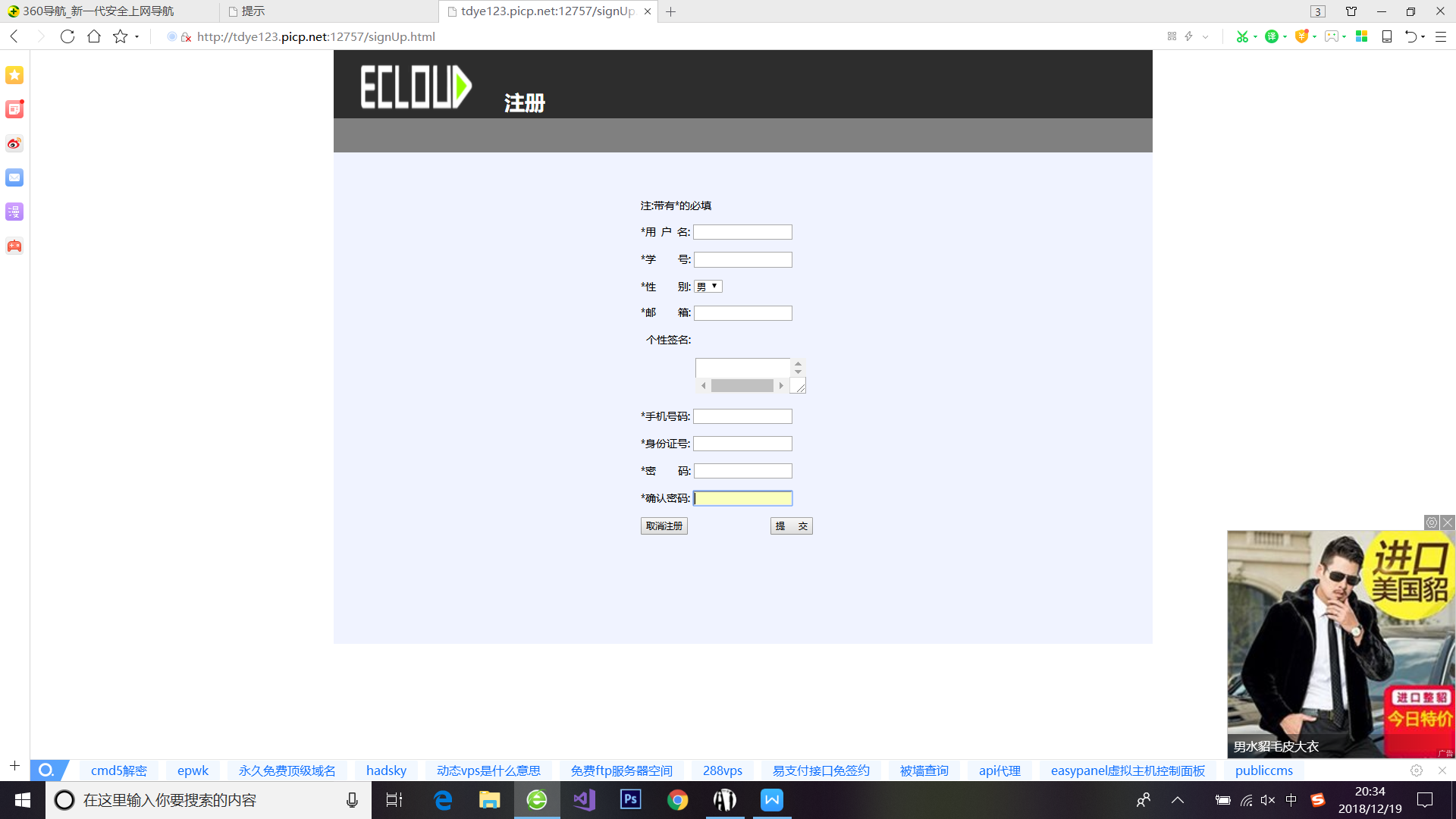
下部分用于填写登录信息。

运用验证码来对用户登录限流。

可以进入注册页面来注册新用户。

下面的密码找回可以通过邮箱来对密码进行修改。

②注册页面



用于注册新用户，设置用户名密码等账户资料，可以检查到不符合要求的输入并弹出警告。

③搜索页面



右上角的用户部分可以查看当前登录账户以及所有积分，也可以进入个人中心和安全退出。下方用于搜索和推荐等，可以根据文件的类别和关键词进行搜索。

④搜索结果



左上仍是个人面板，下方显示根据要求所得出的符合要求的搜索结果，可以看到符合要求的结果数量并进行翻页处理，点击文件的标题可以查看文件的详细资料

左上角显示的是上传用户,可以点击进入个人页面进行登录。

⑤个人中心

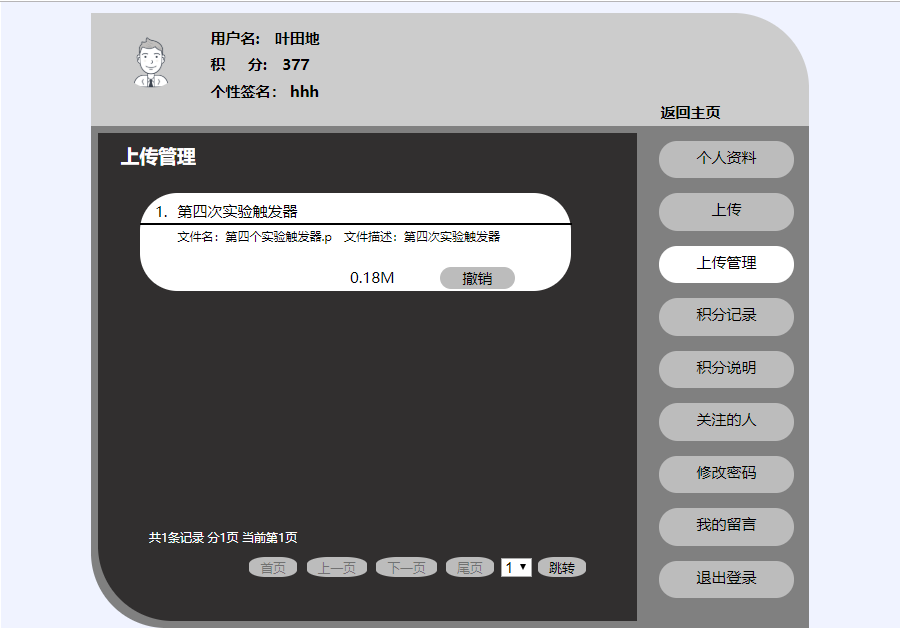
**个人资料：**



**上传**



**上传管理**



**积分记录**

**积分说明：**



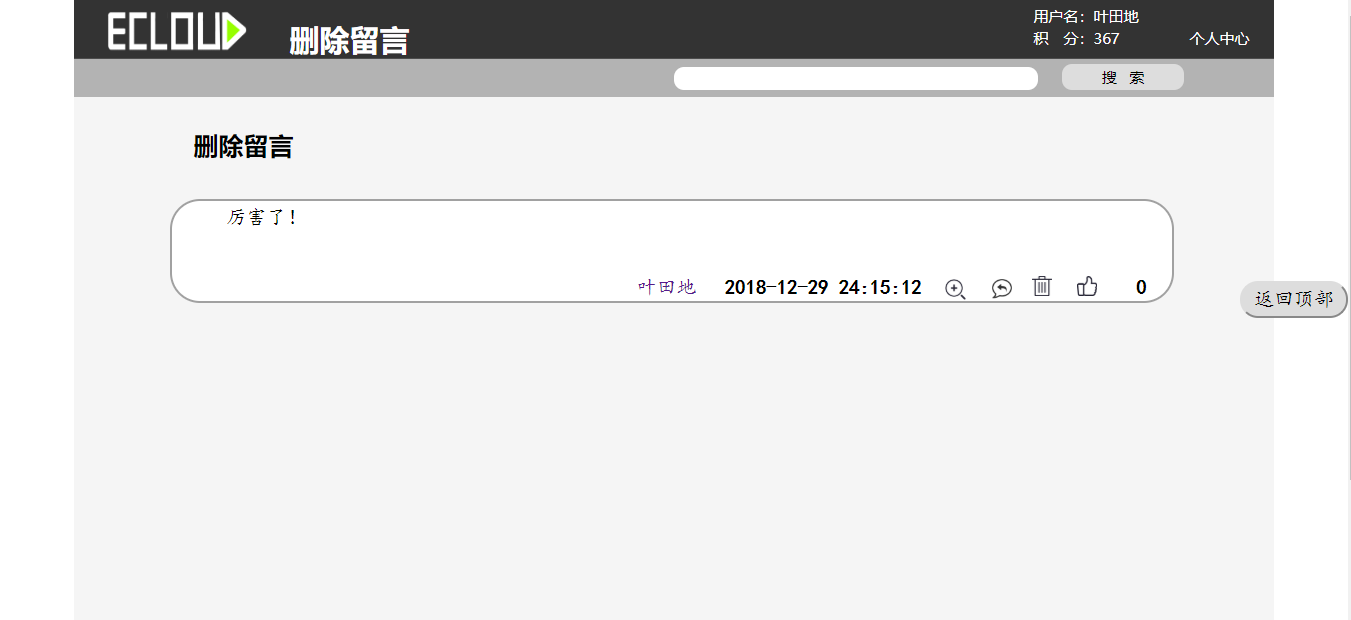
**关注的人**



**修改密码**



**我的留言**



可以对账号进行管理，具体包括：

个人资料的查看与修改；

上传文件，并设置积分标题和描述等信息；

上传管理，对上传的文件进行查看和删除处理；

积分记录，对积分的变化提供记录方便用户查找；

积分说明，了解积分的用处以及获得的方法；

关注的人，用于管理关注的人并方便进入他们的中心；

我的留言，查看自己的留言信息，可对自己的留言进行删除，点赞等操作；

修改密码，修改自己的当前密码，并返回登录页面；

退出登录，返回登录页面。

⑥他人中心



可以查看他人的资料以及上传的文件，也返回自己的中心和退出登录。

⑦文件详情



可以查看文件的资料，对文件进行下载，查看文件的留言,对他人的留言进行回复，点赞等操作，同时可以建立新的留言等。

⑧密码找回



输入自己的电子邮箱后，系统将会向邮箱中发送一个链接，点击链接后便可以重置密码。



点击邮箱中的链接即可进入重置密码页面（服务器邮箱为18365225454@163.com）：





重新输入新密码之后提示密码重置成功，并返回登录页面。

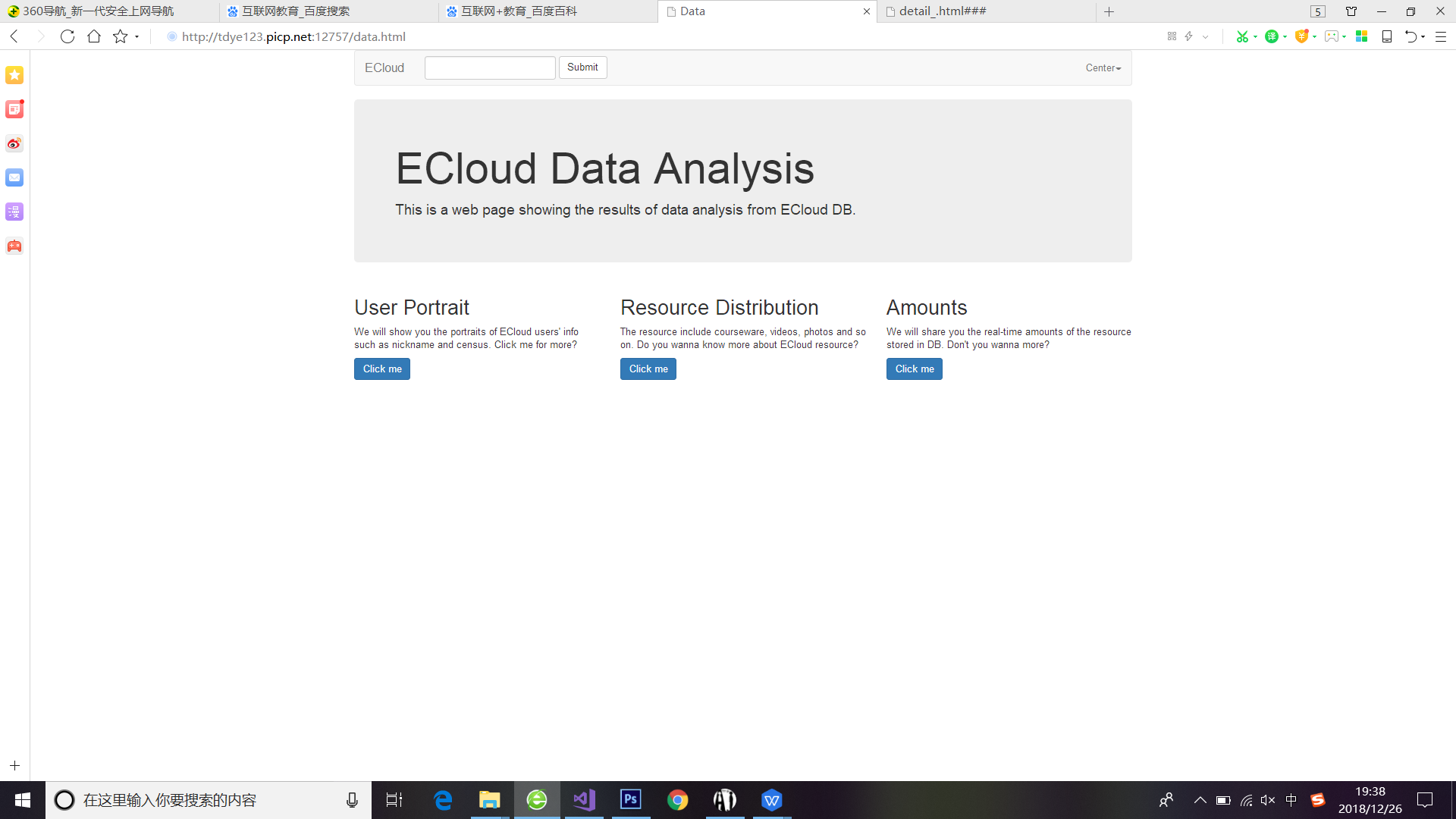


## 4.3管理员界面展示

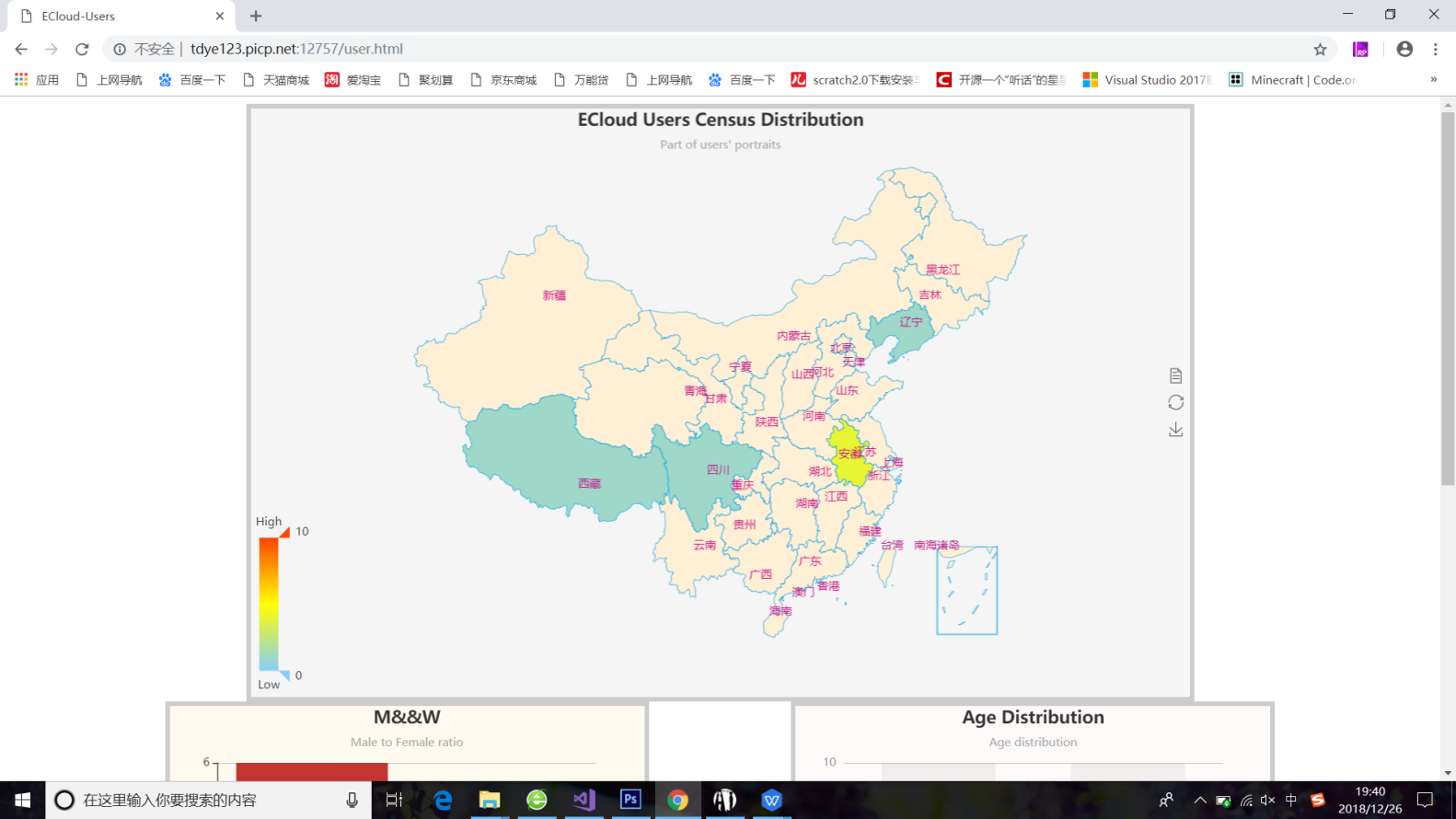
①管理员登录界面

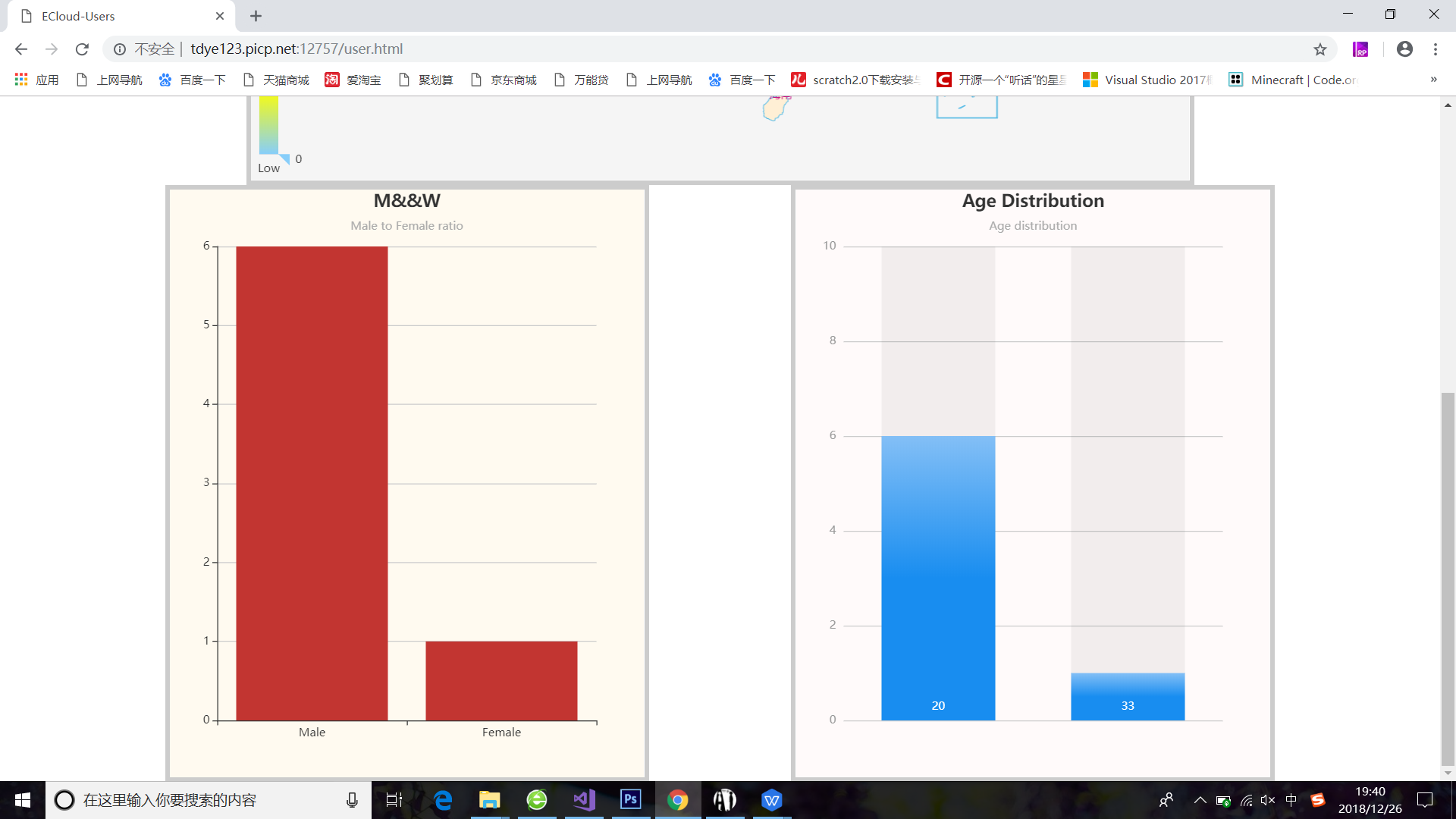


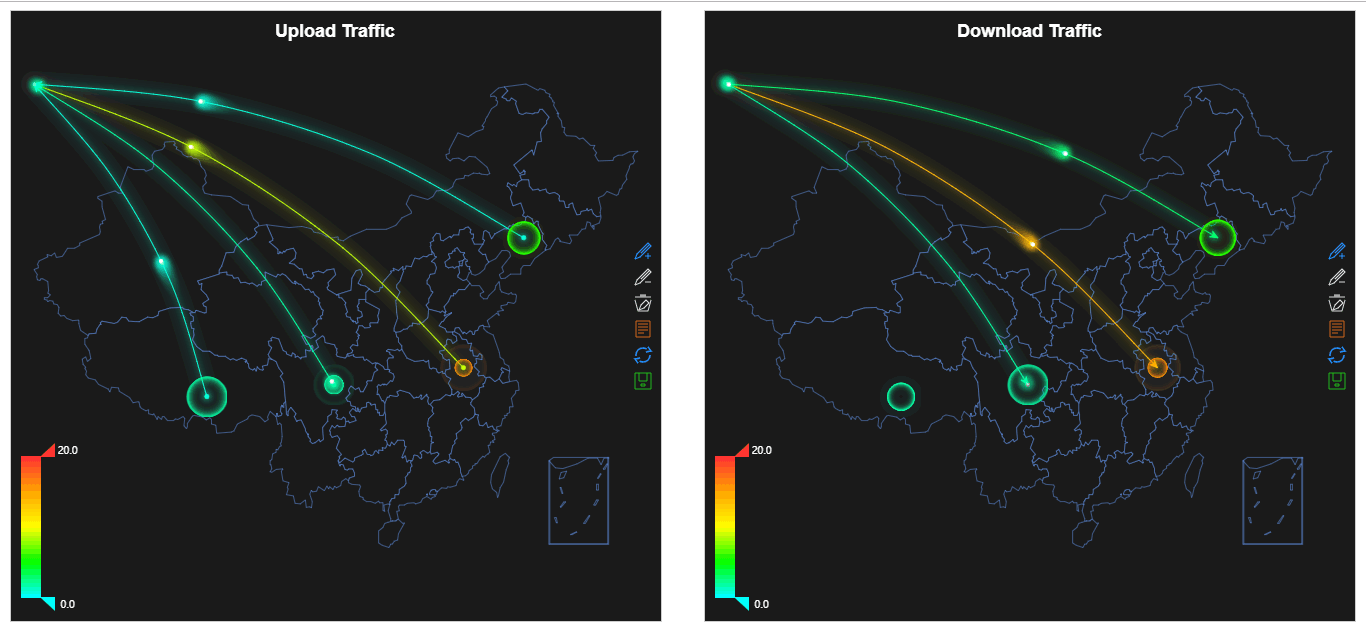
②管理员中心界面



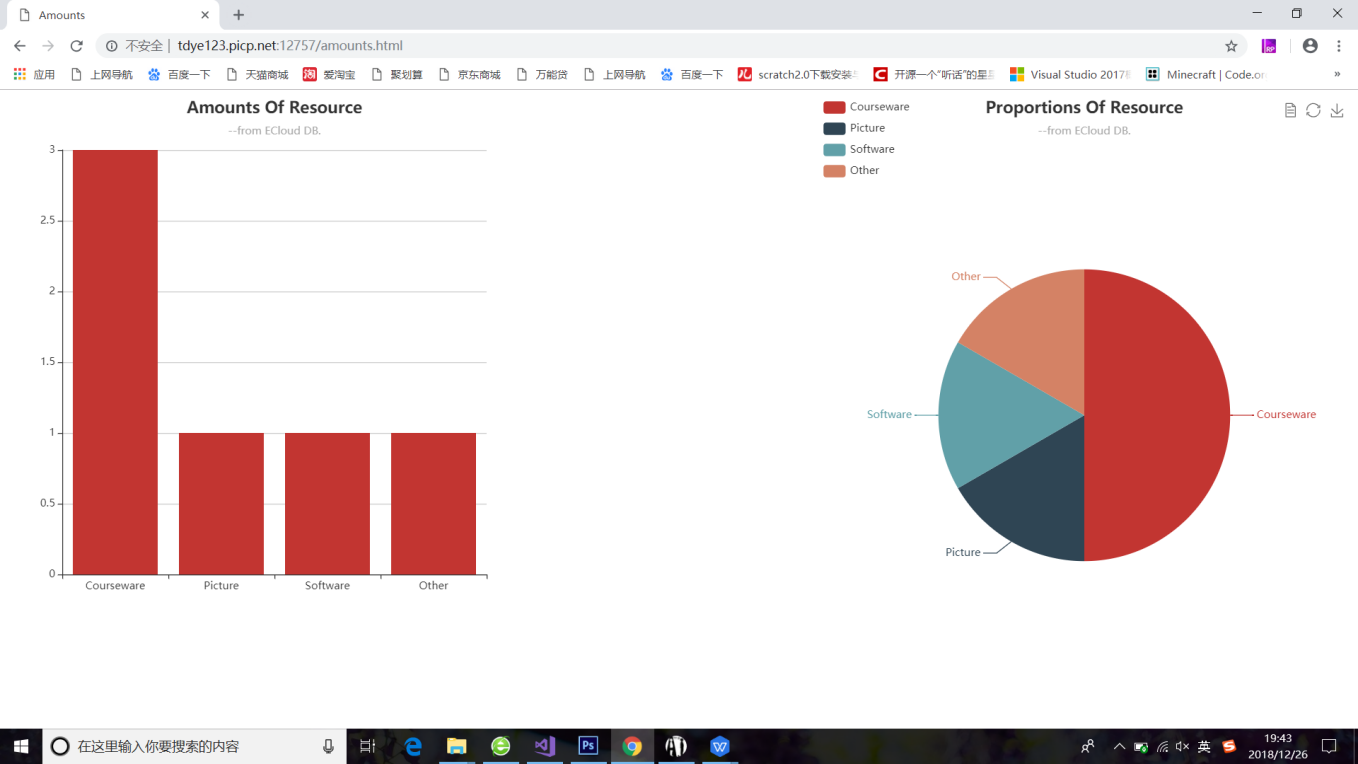
③系统用户地区分布，性别分布和年龄段分布统计





④系统上传下载资源分布统计

⑤系统文件资源数量和类别分布统计



# 5.结论与展望

## 5.1 总结

通过这次的论文工作，经过大量的查阅文献、设计、实现，基本达到了数据库网站设计的预期目标。实现了网站的简单资源检索，资源的上传与下载，留言评论，留言点赞删除，关注他人，基于Neo4j的推荐用户和资源，修改密码等等功能，此外还做了课程学习、资料收集等工作。

根据网络技术发展的现状，本文主要研究了应用于互联网上的各种应用技术、数据库技术以及网站安全技术等，并把它们充分应用于Ecloud文件共享平台的设计与实现中。回顾论文工作主要有以下几条：

1. 全面分析了网站构建中的重要技术，如html、ajax等技术，并对其优劣进行了讨论。
2. 数据库采用了Neo4j和Mysql混用的方式，相对充分发挥二者的各自优势，同时避开Neo4j在读写速度方面的限制。

3）创建网站前，仔细对网站内容进行分析，使网站的结构比较合理，方便以后升级工作。

4）本文采用的浏览器/服务器架构(B/S)作为信息的共享模式，以ODBC技术作为数据库连接方法，以html技术与javascript语言作为网络活动应用的具体开发手段，设计Ecloud文件共享平台。

## 5.2 展望

本次数据库大作业的亮点在于采用了Neo4j数据库和Mysql数据库混用的方式，一方面发挥了Neo4j数据库擅长关系分析的优势，以此来实现推荐功能，另一方面鉴于Neo4j数据库读写速度上的限制，我们将对于关系分析产生较小影响的或与关系分析无关的因素置于Mysql数据库中，间接上提高了整个数据库系统的读写速度。但是，对于Neo4j数据库擅长关系分析这个特点，我们未能将其发挥极致，仅采取了较为普通的推荐算法。希望在后续的应用中，通过获取大量的用户数据进行数据分析，优化推荐算法中的各个权重，使整个系统处于一种学习反馈的状态，该部分我们准备采用TensorFlow深度学习框架实现。

另外，对于留言点赞部分，考虑到信息的易管理性，我们将其置于Mysql数据库中。但是，对于这种对读写速度要求较高的的操作，我们后续工作倾向于将其存储在Redis数据库中，提高整个数据库系统的读写速度。

最后，考虑到整个系统的安全性问题，虽然我们采取了session会话机制，但是还是不免将部分明码暴露在Url地址中，希望后续的学习中可以优化系统的安全性。

参考文献

[1]何建平. 基于B/S模式平台搭建的常见问题的分析与解决[J]. 科技展望,2016,(29).

[2]王小霞. 基于 B/ S 模式数据库检索系统的设计与实现[J]. 微型机与应用,2017,36(6):89-91.

[3]张帜．Neo4j权威指南．北京：清华大学出版社，2017．8-22