

PaperPass检测报告简明打印版

比对结果（相似度）：

总体：5 %（总体相似度是指本地库、互联网的综合比对结果）

本地库：5 %（本地库相似度是指论文与学术期刊、学位论文、会议论文数据库的比对结果）

互联网：0 %（互联网相似度是指论文与互联网资源的比对结果）

编号：575514156527CLZCO

标题：社会化问答网站的设计与开发

作者：周康

长度：14380 字符(不计空格)

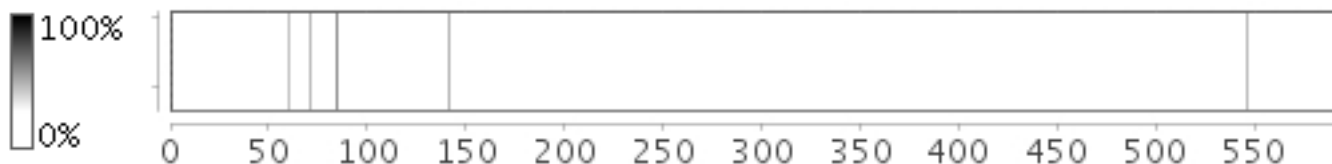
句子数：593句

时间：2016-6-6 14:11:33

比对库：学术期刊、学位论文（硕博库）、会议论文、互联网资源

查真伪：<http://www.paperpass.com/check>

句子相似度分布图：



本地库相似资源列表（学术期刊、学位论文、会议论文）：

1. 相似度：1 % 篇名：《社会化问答网站的知识生产与传播——以知乎网为例进行研究》

来源：学位论文 湖南师范大学 2015 作者：邓海霞

2. 相似度：1 % 篇名：《论垂直型问答网站的发展》

来源：学术期刊 《计算机光盘软件与应用》 2012年21期 作者：刘宏明 柴梓淇

互联网相似资源列表：

没有找到与互联网相似度高的资源！

全文简明报告：

社会化问答网站的设计与开发

摘要：随着科技进步，网络的高速与便捷使社会各个领域的知识以一种惊人的速度传播与发展，人们称21世纪是一个“知识大爆炸”的时代，而一个能够把一些专业知识和经验整合并整理的网站，则是一个非常有用且让无数人收益的知识宝藏。

{ 40 % : 本文重点介绍了一个社会化问答网站的意义以及实现, 网站的主要功能有: } 话题的创建, 问题的创建, 用户参与问题回答, 用户关注话题、问题、答案或者用户, 并在主页可以看到所有有关的状态更新, 用户可以评论和投票答案。 本设计中的每一个问题在创建时都会被要求指定至少一个话题, 这样关注话题的用户就可以在第一时间看到, 用户可以回答问题, 并给答案投票或者评论, 用户可以搜索问题与答案, 并根据答案获得的票数或者答案的评论来确定高质量答案。 本系统基于Django框架, 其中涉及的技术主要是MVT模式的开发技术, 使用Python语言开发, 使用MySQL进行数据管理。

该系统的实现, 旨在帮助用户在知识爆炸的互联网世界里寻找并建议高质量的答案, 介绍一些各行各业的经验, 收集并整理这些知识财富, 共享给每一个使用它的人。

关键词: 问答; 知识; 投票; MySQL; Python; Django; MVT

1 绪论

1.1 课题研究的背景及意义

互联网发展至今, 搜索引擎的出现对整个互联网甚至整个社会都是一个巨大的成就, 搜索引擎方便的帮助人们缩短了寻找资料的时间, 但是同时也带来一个弊端, 那就是当用户去搜索一些模糊性的问题, 搜索引擎通常很难给出满意的答案, 如“xx产品怎么样”这样的问题, 在搜索引擎上只能看到该产品的广告、介绍, 以及可能有这个产品自问自答的网页, 用户很难从中筛选出有用的真正对这个产品的评价。 偶尔用户想寻找某个领域里一些经验性的知识或者资料, 在搜索引擎中可能搜索出一大堆资料, 但是用户没有那么多的精力或时间去从那么多的资料中寻找答案, 这时候如果身边有一个精通该领域的人的帮助, 就会省去很多时间。 而本课题研究的社会化问答网站的作用就是缩短我们与专业人才之间的距离, 让用户可以更轻松的获得我们想要的答案。

社会化问答网站具有打破了原有获取网络信息的模式的优点, 让更有经验的人去用高质量的答案回答问题, { 50 % : 如同创业企业孵化机构 YCombinator 的风投合伙人 Harjeet Taggar 在社会化问答网站 Quora 回答一个问题时表示宁愿从众多高质量的答案中, } 总结出一个正确的答案, 也不愿去通过其他渠道获取有用的信息, 由此可见一个高质量的答案, 可以省去人们更多的时间。

{ 52 % : 对比搜索引擎, 社会化问答网站具有以下几个优点: }

一、强社会影响力。 当一个高质量的回答或者问题引起很多用户的强烈兴趣, 相关的答案或者问题会进入社会化媒体就行二次传播, 具体的传播方法有用户自主传播, 官方推荐或者其他媒体的微博论坛推荐, 其长尾效应给社会带来的影响时间长范围广, 为原始信息传播赋予了新的意义和价值。

二、基于兴趣的强吸引力。 社会化问答网站的使用者可以根据自己的兴趣关注自己喜欢的话题, 既可以积累在这个话题领域里的知识, 有便于平台上知识的传播。

{ 41 % : 社会化问答网站最初的焦点在知识层面, 用户之间联系很弱, 随着时间发展, 关注点相同的用户, } 有着相似的背景和爱好, 用户之间的联系逐渐加强, 最终会发展成一个基于兴趣的社区。 一方面, 增加了用户的黏性, 另一方面, 用户可以通过其他用户提出的兴趣相同的问题学习到更多的知识, 提高信息获取的质量。

三、答案的高质量。 通过搜索引擎的进行搜索, 会有许多垃圾信息的干扰, 虽然现在有百度知道等一类的问题

答类产品出现，但是由于门槛过低，所有用户都可以回答，很容易出现许多为了系统积分而随意写的甚至文不对题的低质量答案，这样的答案并不能满足其他用户渴望专业性较强的回答的需求，而社会问答网站会通过实名制和邀请制来过滤僵尸用户，有效的减少垃圾信息，另一方面，实名制可以很好的让提问者有针对性的向对自己感兴趣的人提问，而回答者出于利益相关或者在某一个领域建立权威形象，会竭尽全力回答问题，内容的品质也可见一斑。

1.2 国内外研究状况

1.2.1. 环境对比

国外的网站以Quora为代表，是国外社会化问答网站的龙头企业，于08年上线，于10年和12年分别融资八千六百万美元和五千万美金。其他国外网站有Aardvark在线问答服务平台，通过直接导入自己的好友，设置感兴趣的话题，获取合适的答案，还有ChaCha，一个为手机端提供问答服务的网站，用户提出问题的方式有多种，通过电话或者短信的方式提问，ChaCha再以短信的方式给用户返回答案。由此可见在功能上，国外的社会化问答网站五花八门。

国内的网站以知乎为代表，是国内最大的社会化问答网站，10年开放，三个月后获得李开复的投资，一年后获千万美元投资，13年面向开放注册，注册用户飙升到400万。其他国内社会化问答网站则有略晓网，是一家付费(有偿)“社会化问答网站”，也是中国第一个付费问答社区，提供网友自助付费咨询、有偿咨询、零技术门槛，{58%：就可以建立一套专属自己的社会化问答网站的平台。}针对开发领域的好问答网站。以及国外问答巨头Quora在中国和嘉伦联合投资创建的国内首个管理咨询类垂直问答系统---即问即答网。

1.2.2. 功能比较

Quora和知乎其主要功能及特点有很大的区别。Quora的主要功能有：积分系统、周边书籍、轻博客。Quora的信用系统是用来计算分数的，每个新成员的基础分为500分，分数用来提问和转赠。如果分数低于100分，会得到系统的新分数，但是和其他网站的分数激励系统不一样的是，Quora的分数不直接与实物挂钩。Quora的周边产品只有一本收藏最近Quora的精彩高质量的回答的书。Quora的轻博客就是和网上的博客类似，用来给用户以博客的方式来分享知识和经验，这是Quora鼓励用户分享知识另一种方式。

{60%：知乎的主要功能有知乎日报、话题广场、发现、专栏、知乎圆桌。}知乎日报是为非知乎会员准备的，用来推送近期热门问答和消息的移动app。话题广场是用来给用户展示所有热门的话题。发现是知乎系统通过算法和编辑的选择，将近期热门的问答置顶。专栏是属于用户自己的栏目，和Quora的轻博客类似，用户可以发表自己的文章。知乎圆桌是知乎在线下举办的活动，邀请知乎大v来参加的互动活动。

1.2.3. 运营管理的特征比较

Quora和知乎的目的都在于创造有价值的内容，但是两家的管理制度不一样。Quora对答案的来源严格把关，如果是从别的地方复制粘贴过来的，需要说明来源，在医药方面尤其谨慎，没有资格证明的医生不能回答医药方面的问题，而且在医药方面，医生只能给出建议，不能开药，最后还要附上免责证明。而这些在知乎只能靠道德约束，目前还没有有效的手段来约束。

1.3 未来的发展趋势

{ 74 % : Stack overflow的创始人Spolsky曾表示在社交和搜索风潮之后,是问答服务的时代。 } { 50 % : 社会化问答网站是一个公共的社交知识平台,他用知识在人与人之间建立了强关系。 } 它通过关注功能和动态更新提示的功能,让用户可以实时了解到最新的某一个领域的知识和最前沿的动态,很简单的第一时间了解到某个领域里面的大神最新分享的经验和知识。

{ 41 % : 搜索引擎面向的是所有互联网用户,所以提供的内容包含各个行业领域,为用户提供与之相关的成千上万的网页, } 但是针对领域里面一些细分的或者冷门的行业,一些对网络依赖不强的行业,与之相关的网页就会非常的少, 提供的内容质量低,针对性也差,而社会化问答网站在此就有针对性强,质量高的优势。

1.4 可行性分析

1.4.1 技术可行性

本系统主要涉及到的技术是Javascript和Python开发技术。 在实习公司中就已经系统的学习了大量相关知识,为网站开发奠定了基础,在公司实习过程中,接触了不少与之相关的项目,有相关开发经验,所以具有技术可行性。

1.4.2 操作可行性

该网站界面采用bootstrap框架,整体结构和布局都十分简洁优雅。 以为兼容性好,所以可以适应不同分辨率的屏幕,以及大部分的主流浏览器,操作方便。 即可以在电脑上使用,也可以在大多数主流手机上运行正常。

1.4.3 经济可行性

本系统用的都是免费软件和框架,开发过程也不需要额外的硬件支持,只需要在电脑上运行本系统即可,所以在经济上,是完全可行的。

1.4.4 方案可行性

通过问答将相同兴趣爱好的用户粘合在一起,在网上已经有知乎、Quora类似的网站的成功案例,所以具有方案可行性。

1.4.5 法律可行性

本系统在开发过程中所用软件都是开源正常软件,网站功能不违反国家相关法律,开发成果完全来自于个人,应当受到国家法律保护。

1.5 课题及论文简介

本系统以Python为开发环境,Django为框架,Mysql作为数据库管理,Bootstrap为网站控制前端样式和结构,Jquery控制前端动作脚本。 在具体设计实现本系统时,考虑并加入了注册和登录功能,以及动态更新的功能。

{ 50 % : 本论文的结构主要包括绪论、需求分析、系统设计、系统实现、用户手册和系统评价六个部分。 }

第一部分绪论主要是描述了网站系统的背景以及意义，国内外状况与发展趋势，并做了可行性分析。

{ 42 % : 第二部分需求分析从三个方面入手，分别是业务流、数据流和数据字典的三个方面分析了系统需求。

}

第三部分系统设计介绍了系统总体的结构设计，接下来以各种图文的方式展示了系统数据模型的设计，最后介绍了一下开发环境的设计。

第四部分系统实现是论文中最核心的内容，介绍了系统所有的模块与功能，将系统的每一个细节通过介绍加逻辑代码展示的方式深刻剖析。

第五部分用户手册介绍了系统的功能特点以及部署方式。

第六部分系统评价说明了一些我对于自己开发的系统的一些想法和评价。

最后还有致谢、毕业论文的参考资料等等内容的介绍。

2需求分析

2.1业务流分析

根据对网站系统的分析，总结出来下面业务流，业务流图如图所示2-1所示。

图 2-1 网站系统业务流图

2.2 数据流分析

本系统的顶级数据流图如图2-2-1所示：

图 2-2-1 顶级数据流图

{ 44 % : 根据系统的顶级数据流图，可以进一步细化一级数据流图，如图2-2-2所示： }

图 2-2-2 一级数据流图

2.3 数据字典

在数据字典里定义的数据表主要有，网站用户信息表，网站动态信息表，话题信息表，问题信息表，答案信息表，评论信息表。 详细描述如下所示：

表2.3.1用户信息表

表2.3.2用户动态信息表

表2.3.3话题信息表

表2.3.4问题信息表

表2.3.6答案信息表

表2.3.5评论信息表

问答系统网站的业务逻辑分析如下：

3系统设计

3.1系统总体结构设计

根据问答系统的功能分析、业务流分析、和数据流分析之后，可以将系统分为六个模块，分别为首页动态、回答问题、关注功能、收藏、搜索和用户界面，具体如图3-1所示：

图3-1 系统总体结构设计图

登录注册注销功能模块： 当用户注册成功或者登录成功以后，后台将当前的 session信息保存到数据库的表中，并在给客户端浏览器返回的数据中种入 cookies， { 48 %： cookies包含着当前会话的 session id，客户端浏览器会接收保存当前 cookies，客户端在每次向服务器发送请求的时候会包含该 cookies， } { 43 %： 服务器后端可以通过客户端发过来的 cookies值比对数据库中的信息来判断用户是否登录，用户注销时， } 则将客户端的 session信息从数据库中删除。

问题归类功能模块： 每一个问题在创建的时候，都要强制选择至少一个所属话题，这样当用户对某一个话题感兴趣的时候， 可以通过搜索话题，找到该话题下所有的问题，方便用户可以了解更多某方面的知识。

点赞反对功能模块： 在每一个问题下的每一个答案，用户都可以点赞或者反对，但是只能点赞或者反对，可以取消点赞和反对， 点赞和反对将会影响问题下答案的排名，答案排名是按照获得的赞的数量从上到下排列。

评论功能模块： 用户可以评论问题下的每一个答案，答案回答者在主页可以看到动态。

关注功能模块： 用户可以关注话题，在某一个话题下有新的问题出现时，用户可以在主页的动态上第一时间看到， 用户还可以关注问题，当某个问题下有新的回答，用户也可以通过主页动态了解到，用户还可以关注答案， 当答案有新的评论时，用户在主页上看到答案有新的动态，用户还可以关注其他用户， 当其他用户有任何点赞反对关注动态，都会在主页看到。

搜索功能模块： 用户在网站任何时候都可以搜索话题、问题或者用户，搜索是用户接触自己感兴趣的內容的主要途径，所以搜索功能的作用，可以让用户很方便的去关注或了解这方面的内容。

3.2系统数据模型设计

3.2.1 数据概念结构设计—E-R图

根据网站系统的数据流分析，系统主要有：话题、问题、答案、评论、用户等实体。每个实体可以用E-R图表示，如图3-2-1所示：

图3-2-1 系统E-R图

3.2.2 数据逻辑结构设计—关系模式

根据上述描写的实体信息和各个实体之间的联系的E-R表，设计了如下所示的实体所包含的各类信息。

用户：

主键： 编号

外键： { 46 %： 关注话题编号，关注问题编号，关注答案编号，关注用户编号，收藏答案编号 }

其他属性： 姓名，是否有效，创建日期

动态信息：

主键： 编号

外键： { 44 %： 事件发起者编号，事件被执行者编号，话题编号，问题编号，答案编号，评论编号 }

其他属性： 事件类型，创建时间

问题：

主键： 编号

外键： 所属话题，创建者编号

其他属性： 问题标题，问题内容，创建时间

话题：

主键： 编号

外键：

其他属性： 话题名，描述

答案：

主键： 编号

外键： 所属问题编号，创建者编号

其他属性： { 48 %：赞同者编号，反对者编号，答案内容，创建时间 }

评论：

主键： 编号

外键： 所属答案编号，所属用户名

其他属性： 评论内容，创建时间

3.2.3 数据逻辑结构设计—关系表

根据上面的关系模式进行数据库设计。

表3.2.3.1用户表

表3.2.3.2动态信息表

表3.2.3.3问题信息表

表3.2.3.4答案信息表

表3.2.3.5评论信息表

表3.2.3.6话题信息表

3.3 开发环境与运行环境设计

3.3.1 开发环境设计

本网站系统采用的Python开发，使用了Django第三方框架，利用了Mysql数据库，对系统的一切操作，都是基于Mysql的基础上执行的。

硬件及软件环境配置：

(1)硬件及操作系统：

操作系统： Windows10旗舰版

计算机品牌：Lenovog480

处理器：Intel(R)_Pentium(R)_CPU_B960_@_2.20GHz

显示芯片：IntelIris 1536MB

主板芯组：IntelHM55

内存容量：2GB

硬盘容量：500GB

(2)应用软件配置：

画图：Microsoft Visio 2010

开发工具：Sublime

数据库：MySQL

开发语言：Python

3.3.2 运行环境设计

要运行本系统需要相关配置如下：

Django 1.6.5

Mysql-Python 1.2.5

Mysql 5.6

4系统实现

4.1网站登录以及注册模块的实现

4.1.1登录功能实现

在用户还未登录系统时，访问网站的任何资源都会自动跳转到登录界面，登录界面由两个表单元素构成，用户名及密码以及登录按钮， { 47 %：当用户名与密码不匹配的时候，系统会自动给出错误提示，并要求重新登陆。

} 主界面如下图4-1-1-1所示：

图4-1-1-1 登录页面

为了实现登录界面展示，需要有一个login.html模板文件，并在当用户点击登录时，用js脚本验证一下两个输入框中是否有内容，如果没有则会给出错误提示。

{ 44 % : 当前端将数据传到服务器时，后端服务器获取到用户名和密码，首先在数据库中的 auth_user表中比对用户名和密码是否正确， } 再进一步判断该用户下的 is_active是否为 True，如果为 True，则表示该用户有效，验证通过， 将当前的 sessionid的值一方面作为 cookies传给客户端浏览器，另一方面，将当前 sessionid值存入数据库的 django_session中， 用以标记该用户登录状态。 客户端浏览器被植入cookies见图4-1-1-2。

图4-1-1-2 客户端植入cookies

代码实现如下所示：

获取用户名、密码;

if (验证 (用户名、密码) 是否为真){

if (用户下属性is_active是否为真){

验证通过，将当前sessionid存入数据库的django_session表中;

将sessionid放到将要返回给客户端消息的cookies值中;

else{

{ 53 % : 将错误消息放入将要返回给客户端的消息; } }}

else{

{ 53 % : 将错误消息放入将要返回给客户端的消息; }

返回消息给客户端;

每一个cookies中的id都唯一，不会发生串号情况。

4.1.2注册功能的实现

登录页面由一个 signup.html做模板，在登录界面基础上多了一个确认密码输入框，并用js脚本控制，当上下密码不一致时，不会发生提交，并给出错误提示，当逻辑正确时， { 44 % : 数据会提交到后台，后台首先验证当前用户名是否存在，如果存在，就返回错误信息， } { 45 % : 如果正确，则在数据库中添加用户信息，并执行登录过程，登录到主页。 } 注册界面如图4-1-2所示：

图4-1-2 注册界面

代码实现如下所示：

获取用户名、密码;

if (用户名 in users表为真){

返回 “ 用户已存在 ” 错误信息;

else{

在数据库中添加用户信息;

执行登录过程;

4.1.3注销功能的实现

相比较其他功能而言，注销功能就简单的多，只需要将当前sessionid在数据库中删掉，同时返回一个重定向给客户端浏览器即可。 注销功能在网站登录后右上角个人模块中，如图4-1-3所示：

图4-1-3 注销功能

4.2搜索功能模块的实现

搜索功能是网站最重要的一项功能之一，通过搜索功能，用户可以到达想去的任何地方，本系统的搜索功能放在了网站的顶部，这样在任何时候都可以搜索自己想要的内容。 搜索框见图4-2-1：

图4-2-1 搜索框

搜索框旁边有一个下拉框，一共有三个选项可以选，分别是搜索话题、搜索问题、搜索用户，当用户搜索内容为空的时候， 页面不会跳转，当用户输入想搜的内容时，系统会根据给出的内容进行模糊搜索，匹配出所有的结果， 返回给客户端浏览器，图4-2-2就是搜索用户“ ad ”，系统给出的结果：

图4-2-2 搜索用户“ ad ”

用户搜索时候，前台浏览器会发送两个数据给系统后台，一个是搜索类型，另一个是搜索内容，后台拿到数据以后， 首先判断类型，如果是搜索话题，则在 qa_theme表下的 theme_name值中找涵盖搜索内容的主题， 找到后返回他们的完整 id与名字，在循环每一个获取到的主题列表中的 id，判断该主题的 id是否在用户的关注列表中，
{ 42 %： 将这些值组合到一起存为一个大列表，返回给客户端。 }

搜索问题的方式同话题差不多，但是搜索用户时会不一样，因为我觉得用户自己是不可以关注自己的， 但是可以搜索到自己，所以后台在搜索用户时，如果结果数据中包含自己的话， 将在结果页面的右侧不会显示关注或者取消关注的按钮。 如图4-2-3所示：

图4-2-3 搜索结果包含自己的时候不会有关注按钮

代码实现如下所示：

获取搜索类型和搜索内容;

```
if (搜索类型== ' theme ' ){
```

结果列表=查询数据库qa_theme中theme_name字段中包含搜索内容的结果;

```
for (结果in 结果列表){
```

```
if (结果.id in 当前用户关注列表里){
```

```
    结果.focused=true;
```

```
else{
```

```
    结果.focused=false;  }}
```

```
返回结果}
```

```
if (搜索类型== ' question ' ){
```

结果列表=查询数据库qa_question中question_name字段中包含搜索内容的结果;

```
for (结果 in 结果列表){
```

```
if (结果.id in 当前用户关注列表里){
```

```
    结果.focused=true;
```

```
else{
```

```
    结果.focused=false;  }}
```

```
返回结果}
```

```
if (搜索类型== ' user ' ){
```

结果列表=查询数据库user_profile中username字段中包含搜索内容的结果;

```
for (结果 in 结果列表){
```

```
if (结果.id in 当前用户关注列表里){
```

```
结果.focused=true;

else{

结果.focused=false;

if (结果.id == 当前用户id){

结果.display=true;

else{

结果.display=false; }}

返回结果}

{ 40 % : 当搜索结果为空时，页面也会正确显示，见图4-2-4： }
```

图4-2-4 搜索结果为空时

4.3 关注功能的实现

关注功能是系统的一个重要功能，其主要作用是给用户带来某一领域的话题问题等各种动态的更新，是将网站从一个用户主动去搜索寻找资源的过程， { 48 % : 变成了一个网站主动将最新的状态，第一时间推送给用户的过程。 }

关注功能并没有一个独立的页面，而是散布在各个页面中，用户一共可以关注四个类型：话题、问题、答案、用户。

4.3.1 关注话题

用户关注话题，当话题下有新的问题出现时，用户可以在主页第一时间看到更新。

关注话题如图4-3-1-1所示

图4-3-1-1 关注话题按钮

关注按钮在搜索时也会出现，如图4-3-1-2所示：

图4-3-1-2 搜索话题

关注按钮通过js控制监听，当点击按钮时，前台就会发送ajax数据给后台，数据内容包含关注类型和关注id两个字段。

后台拿到数据，就会根据关注类型执行相应代码，将关注者id放入当前用户的focused_theme_ids字段中。

代码实现如下所示：

获取数据;

```
if (关注类型== ' theme ' ){  
  
if (theme_id in 用户. focused_theme_ids){  
  
用户. focused_theme_ids+= theme_id;  
  
else{  
  
用户. focused_theme_ids-= theme_id;
```

4.3.2 关注问题

{ 41 % : 用户关注问题与关注话题的逻辑相似，关注问题见图4-3-2 : }

图4-3-2 关注问题按钮

与话题不同的是，当问题下有新的答案时，用户可以在主页第一时间知道。

代码实现如下所示：

获取数据;

```
if (关注类型== ' question ' ){  
  
if (question_id in 用户. focused_question_ids){  
  
用户. focused_question_ids+= question_id;  
  
else{  
  
用户. focused_question_ids-= question_id;
```

4.3.3 关注答案

{ 41 % : 用户关注答案与关注话题的逻辑相似，关注问题见图4-3-3 : }

图4-3-3 关注答案按钮

与话题不同的是，当答案被点赞、反对或者评论时，用户可以在主页第一时间知道。

代码实现如下所示：

获取数据;

```
if (关注类型 == 'answer') {  
  
    if (answer_id in 用户. focused_answer_ids) {  
  
        用户. focused_answer_ids += answer_id;  
  
    } else {  
  
        用户. focused_answer_ids -= answer_id;  
    }  
}
```

4.3.4 关注用户

{ 45 % : 用户关注用户与关注话题的逻辑相似，关注问题见图4-3-4 : }

图4-3-3 关注答案按钮

与话题不同的是，当被关注的用户有任何动态时，用户可以在主页第一时间知道。

代码实现如下所示：

获取数据;

```
if (关注类型 == 'user') {  
  
    if (user_id in 用户. focused_user_ids) {  
  
        用户. focused_user_ids += user_id;  
  
    } else {  
  
        用户. focused_user_ids -= user_id;  
    }  
}
```

4.4 点赞和反对功能的实现

点赞与反对功能是为了激励用户写出更好的答案而设计的一项功能，一个好的答案，往往能获取很多用户的支持。 { 40 % : 而每个用户对于每一个答案都会有一个投票的权利，即赞成或者反对。 } 如图4-4：

图4-4 赞成与反对按钮

前台浏览器通过 js 脚本实现与后台 ajax 通信，即监听赞同与反对的按钮的点击事件，当用户点击任何一个按钮时，前台浏览器就会用 ajax 将当前用户的所在答案的 answer_id 与点赞或反对的事件类型传到后台，后台拿到数据之后，就会找到在数据表 qa_answer 中对应的 answer 数据，在根据传给后台的事件类型将当前用户 id 加入 up_owner_user_ids 或者 down_owner_user_ids 中，在收集当前答案的赞的数量和反对的数量，用 json 字符串方式返回给客户端服务器， { 42 % : 客户端浏览器接收到数据，重新渲染对应点赞和反对数据。 }

代码实现如下所示：

获取数据;

answer=数据表qa_answer中id与answer_id相同的数据;

```
if (action_type == ' 点赞 ' ){
```

```
if (当前用户id in answer. up_owner_user_ids){
```

```
answer. up_owner_user_ids+=(当前用户id)}
```

```
if (当前用户id in answer. down_owner_user_ids){
```

```
answer. down_owner_user_ids+=(当前用户id)}
```

```
else{
```

```
if (当前用户id in answer. up_owner_user_ids){
```

```
answer. up_owner_user_ids+=(当前用户id)}
```

```
if (当前用户id in answer. down_owner_user_ids){
```

```
answer. down_owner_user_ids+=(当前用户id)}
```

由上面一段逻辑代码可知，当用户点击点赞按钮时，点赞成功，当再次点赞就会取消赞，另一方面，当用户已点赞时候，再去点击反对按钮，就会自动取消当前的赞，逻辑上实现了用户对于每一个答案只有一次投票权的功能。

4.5 收藏功能的实现

收藏功能是方便用户的一项小功能，当用户看到有用的答案时，就可以收藏它，然后再个人中心可以看到收藏过的答案。收藏功能按钮见图4-5-1，收藏结果见图4-5-2：

图4-5-1 收藏答案按钮

图4-5-2 收藏结果

主要代码如下所示:

收集数据

```
if (answer_id in 当前用户.collected_answer){
```

```
    当前用户.collected_answer-= answer_id}
```

```
else{
```

```
    当前用户.collected_answer+= answer_id}
```

4.6 添加问题功能的实现

添加问题是系统重要的基础功能之一，通过各种各样的问题，网站内容得以丰富。用户在添加问题时默认关注添加成功的新问题，这样当问题下有新的答案时就会在第一时间看到。见图4-6-1为添加问题页面：

图4-6-1 添加问题页面

问题标题与话题为必填项，话题输入框为搜索框，主要是为了方便给问题加上需要的话题标签，如图4-6-2为搜索话题是出现下拉框，4-6-3为点击其中一个选项出现的结果。

图4-6-2 搜索话题出现下拉框

图4-6-3 点击其中一个话题的效果

话题搜索的逻辑代码与前面4.2节搜索功能实现相似。

当用户填好添加问题的输入框提交之后，前端脚本会先验证数据是否正确，如果不准确则弹出错误提示。

后台接受三组数据，问题标题，所属话题，以及问题描述，在数据库的 qa_question 中创建一条数据，同时，如果问题所属多个话题，则在动态表中循环话题列表，创建动态信息。

主要代码如下所示：

收集数据;

question=在qa_question表中创建一条数据;

当前用户.focused_question_ids+=question_id;

```
for (话题 in 话题列表){
```

动态表创建(话题)}

4.7 添加答案功能的实现

当用户在某一个问题下添加答案时，整个问题的答案列表会更新，同时用户会默认关注自己创建的答案，当用户的答案被赞的人数多，会被顶到答案列表的第一位。 用户添加答案界面如图4-7-1所示：

图4-7-1 用户添加答案

前台通过js脚本将当前问题id与答案内容传给后台，后台接受数据在数据表中创建一条答案数据。

主要代码如下所示：

后台收集数据

answer = 创建一条qa_answer数据(当前问题id，答案内容)

当前用户.focused_answer_ids+=当前问题id

动态表创建一条新数据

返回answer.id

4.8 添加评论功能的实现

用户评论是用户与回答者互动的最好的方式，通过给答案评论，回答者可以在首页动态了解到有用户对他的答案有了新的想法，从而与评论者互动。 添加评论功能演示见图4-8：

图4-8 测试评论

前端通过js脚本收集评论的答案编号和评论内容，发送给后端，后端接受到数据在qa_comment创建一条数据，创建成功后返回成功信息。

主要代码如下所示：

收集数据

在qa_comment创建一条数据

返回成功信息

5用户手册

5.1功能简介

本系统的作用是希望建立一个良好的社会化问答网站，让所有用户参与分享自己的知识与经验。

用户可以在“添加问题”界面提出问题，通过选择话题，能够让所有共同关注同一个话题领域的人看到，这其中不乏有处理过或者遇到过与提问者相同问题的人，通过他们的经验指导和答案质量以及其他用户投票数，用户可以获得自己满意的答案。

用户在主页的动态更新中可以看到自己关注的话题下的最新问题，选择自己感兴趣的问题回答，答案的质量越高就会获得越多的赞同，回答着可以收获认同感，能得到某一领域里其他人的认可甚至成为权威。

用户可以给答案评论，回答者可以在动态上看到更新，并与之参与讨论，就某一个方面继续探讨和延伸。用户可以觉得有用的答案，可以收藏他们，以备以后再次查看。

用户可以给答案点赞或者反对，点赞的数量将会影响到答案的排名。

用户可以关注话题、问题、答案与用户，当其中的任何一项发生改变时，用户可以通过主页动态获取到最新消息。

5.2系统部署

首先部署环境需要安装python的2.7版本，可以通过官网去下载，本地需要配置好Mysql，然后再通过在<https://pypi.python.org/pypi>中搜索MySQL-Python，下载安装1.2.5版本，再搜索Django下载安装1.6.5版本，至此环境配置完毕。

如图5-2-1所示，系统配置是保存在QASite文件夹下的settings.py文件里，默认系统使用的数据库名为questionanswer，默认数据库用户名与密码都是test，在部署的时候可以任意更改，只要部署环境中存在该数据库和用户即可。

图5-2-1 系统文件结构

首先双击项目下的init_server.bat文件，执行系统的初始化工作，如果执行过程中出现错误，如图5-2-2所示：

图5-2-2 错误提示

{ 41 % : 请检查settings.py文件里数据库配置信息填的是否正确。 }

系统初始化完成之后再双击start_server.bat文件，服务器启动如图5-2-3时，表示系统启动成功。

图5-2-3 启动成功

到这一步系统的部署已经完成，打开浏览器访问<http://127.0.0.1:8000/>即可看到本网站系统。

6系统评价

6.1 系统的特色

本文设计和实现的社会化问答网站目的在于知识与经验传播，受众是那些不满足于从搜索引擎获取到知识的人。用户可以随时搜索自己感兴趣的内容，获取最专业和优质的知识和经验，同时也可以提出问题，与所有有共同兴趣的人一起共享和探讨遇到的问题，真正做到了一个良性的循环。本系统主要为用户提供了以下一些功能：

- (1) 创建话题、创建问题、创建答案、添加评论。
- (2) 点赞、反对和收藏答案。
- (3) 关注话题、关注问题、关注答案、关注用户，并在主页看到所有的动态信息。
- (4) 个人中心查看所有收藏过的和关注过的内容。
- (5) 搜索框可以搜索话题、问题与用户，可以通过搜索到达网站的任何地方。

这些功能组合到一起真正实现了一个社会化问答网站系统所需要的所有基础功能。

6.2 系统存在的不足与改进方案

社会化问答网站虽然实现了基本的主要功能，但在整体表现上还是有不足：

- (1) 在用户开始接触本系统时应该有一个新手教程一样的入门推荐，让用户可以选择自己感兴趣的话题，不然小白用户在刚进系统时，以为没关注任何内容，面对空白的主页动态会不知所措。
- (2) 网站整体界面虽然简洁大方，但是网站整体在样式上显得单调，在内容上会给人一种不丰富的感觉。
- (3) 由于时间精力有限，系统与其他问答网站相比，缺少了一些人性化的小细节功能，比如查看一个答案哪些人给该答案点了赞或者反对。同时注册和登录界面没有设置验证码功能，会有被其他人利用爬虫做一些类似刷赞活动的风险。

结论

{ 43 % : 本网站系统是基于Python的社会化问答网站，是用户之间传播知识与经验的平台。 } 使用本系统的用户，可以轻松的关注自己感兴趣的所有话题，提出问题并与其他同样感兴趣的用户一起探讨问题与答案。

本文研究的主要内容包括：

- (1) 基于cookies实现登录的功能。 { 43 % : 由于 http 是一种无状态的协议，在实现登录功能上，需要人为的去实现，即前台发出请求， } { 46 % : 后台接收用户名和密码，经过与数据库比对，正确则将当前会话的 session 信息存到数据库中， } 同时在返回的信息中包含 cookies，这样浏览器在后面的数据请求中都会有此 cookies，后台通过此 cookies 比对数据库中的登录信息，如果正确则表示用户已登录。这样实现了一个系统的登录功能。

(2) 数据的传递与存取。对于用户来说，网站是一个可视化的存在的系统，但是对于服务器来说，网站系统是一个接收请求数据，结果运算和数据库通信，将一段计算后的文本返回给客户端浏览器的过程，对于后台服务器来说，{42%：后台所做的工作只有两种类型，与数据库通信和计算。}通过客户端浏览器发送的请求，在数据库中增删改查，将取到的数据经过计算，将数据返回给模板，后台再将关键的数据通过在模板中查找，替换关键词，最后将计算好的模板以文本的方式返回给浏览器，一段请求的后台处理完成了。

致谢

终于完成了程序与论文，心中松了一口气，在完成的过程中遇到了很多问题，所幸的是在仲老师的帮助下都克服了。仲老师在我的选题、需求分析、系统设计、系统实现的过程中给予了细心的指导，在程序设计与开发的过程中给予了可靠的建议和帮助，在我遇到不懂的问题时，仲老师也非常尽责而且有耐心的为我解说。老师无私的帮助和教诲让我受益匪浅，也让我对一个程序设计过程有了更充分和详细的认知。

同时，我也要感谢学院领导提供的便利的学习环境，在安静的图书馆，我可以认真专心致志的完成我的论文与系统，不被外界打扰，在需要查找资料时也能很轻松的发现想要的资料，便利轻松的学习环境对我完成论文与系统的帮助很大。

此外，我还要感谢宿舍的各位室友在大学四年中对我的帮助，不在校的时候，学校很多事务都是他们帮忙处理的，如果没有他们，我会多出许多麻烦并且浪费很多时间，多亏他们的帮助，让我能够腾出时间去完成我的毕业设计。

还要感谢最后答辩的各位老师，提出各种宝贵的意见，让我的程序和论文更加的完美。

参考文献

[1] Gourley等. HTTP权威指南. 人民邮电出版社，2012

[2] The Web framework for perfectionists with deadlines | Django. <https://www.djangoproject.com/>

[3] The Django Book. <http://djangobook.py3k.cn/2.0/>

[4] JavaScript 参考文档. <https://developer.mozilla.org/zh-CN/docs/Web/JavaScript>

/Reference

[5] (美)弗拉纳根.Javascript权威指南(第6版). 机械工业出版社，2007

[6] (美)(Nicholas C.Zakas)扎卡斯.JavaScript高级程序设计(第3版). 人民邮电出版社，2006

[7] Bootstrap 中文文档. <http://v3.bootcss.com/>

[8] (美)丘恩(Chun, W.J.).Python核心编程.人民邮电出版社，2008

[9] w3school 在线教程.[http: //www.w3school.com.cn/](http://www.w3school.com.cn/)

[10] (美)Cesar Otero. jQuery高级编程 , 2013

检测报告由PaperPass文献相似度检测系统生成
Copyright 2007-2016 PaperPass