**北京理工大学远程教育学院**

**毕业设计（论文）**

**题 目：个人博客**

专 业：

教学站 ：

学生姓名 ：

指导教师 ：

日 期 **： 2019313**

摘 要

最近几年博客在国内发展的非常迅猛，它可以作为其使用者个人或者工作中的文章、日记、照片等分享的平台，其渐渐成为新兴的网路媒介，并且扩展至营销、商业推广等商业应用，越来越多的人们建立了属于自己的博客来展示自己的风采。

该个人博客站设计的目的旨在建立以Node.js+MongoDB数据库为工具，功能简洁、结构灵活且精致、轻巧的个人个人博客站系统。其中网络日志的管理作为本系统的主要目标，且同时可以满足用户对于图片，相册，留言，个人信息等的管理和展示，并且可以设置博客的背景音乐等功能。

该设计的开发环境为Node.js，使用Node.js编程语言进行编码，采用的是MongoDB数据库。

**关键词：**Node.js； 个人博客； 数据库； MongoDB

目 录

[摘 要 2](#_Toc5048614)

[目 录 3](#_Toc5048615)

[前 言 1](#_Toc5048616)

[第一章 系统概述 3](#_Toc5048617)

[1.1研究背景 3](#_Toc5048618)

[1.2研究现状 3](#_Toc5048619)

[1.3设计内容 4](#_Toc5048620)

[1.4 课题研究意义 4](#_Toc5048621)

[1.4 开发的目标 5](#_Toc5048622)

[第二章 系统开发环境 6](#_Toc5048623)

[2.1 MongoDB数据库 6](#_Toc5048624)

[2.2 Node.js简介 6](#_Toc5048625)

[2.3 Bootstrap 7](#_Toc5048626)

[第三章 需求分析 8](#_Toc5048627)

[3.1 可行性分析 8](#_Toc5048628)

[3.1.1技术可行性 8](#_Toc5048629)

[3.1.2经济可行性 8](#_Toc5048630)

[3.1.3操作可行性 8](#_Toc5048631)

[3.1.4法律可行性 8](#_Toc5048632)

[3.2 功能需求 8](#_Toc5048633)

[3.3 性能需求 9](#_Toc5048634)

[第四章 系统设计 11](#_Toc5048635)

[4.1概述 11](#_Toc5048636)

[4.2系统结构 11](#_Toc5048637)

[4.2.1技术选型： 11](#_Toc5048638)

[4.2.2功能： 11](#_Toc5048639)

[4.3 数据库设计 12](#_Toc5048640)

[4.3.1 E-R 图 13](#_Toc5048641)

[4.3.2 数据库结构 14](#_Toc5048642)

[第五章 系统实现 16](#_Toc5048643)

[5.1用户登录模块的实现 16](#_Toc5048644)

[5.2博客首页 18](#_Toc5048645)

[5.3博客留言板 21](#_Toc5048646)

[5.4博客管理后台（写博客页面） 22](#_Toc5048647)

[5.5博客后台管理页面（文章管理） 25](#_Toc5048648)

[5.6博客后台管理页面（用户数据） 26](#_Toc5048649)

[5.7其他关键代码实现 26](#_Toc5048650)

[5.7.1引用文件 26](#_Toc5048651)

[5.7.2注册组件 27](#_Toc5048652)

[5.7.3获取用户地理位置组件 29](#_Toc5048653)

[5.7.4分类组件 31](#_Toc5048654)

[5.7.5增删改查基本操作 32](#_Toc5048655)

[第六章 总结 37](#_Toc5048656)

[致 谢 39](#_Toc5048657)

[参考文献 40](#_Toc5048658)

前 言

个人博客管理系统是一个给人们提供抒发个人情感、提供人与人之间进行良好沟通的平台。博客拥有真实的内容，人们可以通过博客记录下工作、学习、生活和娱乐的点滴，发表文章和评论，从而在网上建立一个完全属于自己的个人天地，供朋友们浏览和分享，是当今网络最为个性化和平民化的个人展示空间。对于博客用户，只要拥有博客，拥有不同以往的全新网上生活；对于博客服务商，则必须从功能提供转型到全方位社会服务的提供，建立虚拟社会，并负责维护运行，保证博客日常生活，所以开发一个简易的个人博客系统就显得尤为重要。

传统电子文章属于人类传播以及存储，也属于分析、处理和传送信息的主要途径之一。个人博客中最为关键的组成部分之一就是文章信息的管理，因而这个系统的实现就很具有意义性。随着个人博客的不断扩充，例如新浪博客，已由当初的20万余万人至现在的4600万余人，这足以说明系统的必要性，而且阅读的人群也在逐年上涨，人们对于知识的渴求已经不再是普通的打发时间，而是上升至精神方面的需求。可是这些需求仅靠人工操作是绝对不够的，如果今天刚录入1万余篇文章，明天再近2万余篇文章，那么个人博客的管理人员的工作量会大大提高。当然大部分个人博客管理人员是不太精通计算机的使用，使得他们对这个现代化的管理化系统充满抵触，这便加大了实施的难度，所以推出的这个系统需要更加方便易学，更加易于操作，更加方便迅速。为了解决这一技术难题，需要结合现代的网络技术，将博客信息数字化，设施集成化，创建一个自动化系统，使得个人博客案在教学和科研工作上发挥重要的作用。

2000年后，我国个人博客系统的研发曾经出现了一个高潮时期，当时推出了目前在国内较有影响的几个大型自动化系统（如新浪博客、腾讯博客、搜狐博客），并提出了“第三代个人博客系统”的概念。个人博客管理系统是典型的信息管理系统(MIS)，其开发主要包括后台数据库的建立和维护以及前端应用程序的开发两个方面[1]。对于前者要求建立起数据一致性和完整性强.数据安全性好的库。而对于后者则要求应用程序功能完备，易使用等特点。因此，开发这样一套管理软件成为很有必要的事情，在下面的各章中本文将以开发一套个人博客管理系统为例，分析开发过程和所涉及到的问题及解决方法。

经过分析如此情况，使用开发工具，利用其提供的各种面向对象的开发工具，尤其是数据窗口这一能方便而简洁操纵数据库的智能化对象，首先在短时间内建立系统应用原型，然后，对初始原型系统进行需求迭代，不断修正和改进，直到形成用户满意的可行系统。

本文主要从个人博客管理系统研发现状入手，详细的论述了本次设计的开发意义和开发目标，阐述了MongoDB数据库、Node.js以及服务器的具体含义，并在系统需求分析的基础上，对个人博客管理系统进行系统的设计，最终对网站前台模块和网站后台模块进行测试。

**第一章 系统概述**

**1.1 研究背景**

个人博客主页是对未曾见面之人的主要信息来源地，现代化信息技术的迅猛发展，大大丰富了各种形式的信息资源，同时各类个人博客资源量也日益巨大，个人博客的管理工作已经全面进入了信息时代。面对众多的博客和数字化信息，需要对它们进行科学的管理，方便广大网友的使用，要求结合现代化的手段，开发出符合个人博客实际需求的系统来完成对个人博客内各类资源的高效管理[2]。个人博客管理系统实现了对个人博客页面内的各类资源以及信息进行规范化的管理，完成对个人博客馆正常流通业务中产生的数据信息的管理，包括采购个人博客信息、个人博客管理信息等。面对每天的个人博客流动量，个人博客管理系统需要进行海量的数据处理。为适应时代需求，设计和实现一个使高校个人博客的管理更加规范化、科学化和合理化的个人博客管理系统是现在亟待解决的问题。高效合理的个人博客管理系统，对于平台管理者来说，将其从复杂繁琐的数据记录、核查及统计的工作中解放了出来，简化了工作流程，降低了数据出错率，提高了工作效率。对于使用者来说，减少了检索博客的时间，同时合理的个人博客推荐，有利于读者对同类博客的比较阅读，提高了使用体验。因而，研究和实现个人博客管理系统具有重大的现实意义。

**1.2 研究现状**

近几年来，国内外与个人博客管理相似的有关的系统和程序的发展都非常迅速，众多的软件有如雨后春笋般地不断涌现。这些软件尽管各有特点，但从功能上来分析，大致可分为如下两种：基于web的Internet/Intranet模式的个人博客管理程序，如各个国家的国家级个人博客馆，大部分都建成了这种类型的个人博客管理系统，供公众使用。公众号性质的博客管理系统，可以采用基于web的程序，也可采用单机程序[3]。

个人博客管理信息系统在我国已经得到了应用，其效果是明显的。一方面是取得了可计量的直接效果，如节省了人力，减少了费用等；另一方面取得的效果是无形的，是无法直接计量的。如提高了后台管理人员的素质，促进了管理的现代化等。在个人博客管理系统中，一切的设计理念都是为了实现个人博客的高效率和准确性。系统还具备个人博客的查询管理并提供进行相关信息的统计分析[4]。

**1.3 设计内容**

网站的界面很友好、明快、舒适；操作起来容易、快捷；查询资料快捷、准确。个人博客站在很大程度上实现登入的简易性、操作的简单性与维护的最小化。系统运行有较高的稳定性、安全性和可靠性。在功能上游客身份可以用不注册，直接通过匿名访问的方式在博客前台进行查看用户的博客日志，发表留言，查看用户的相册图片等操作。博客用户可以在前台的页面登录，然后进入后台页面。在后台可以进行发表日志和管理日志以及日志的留言，添加相册，删除相册，上传、删除图片等操作。博客的主人在登录之后可以修改自己的个人信息，也可以设置自己博客的背景音乐。

**1.4 课题研究意义**

一个出色的个人博客管理系统能够让网友很快找到信息，并且便于操作，实现个人博客的规范化管理。将所有的个人博客资料集中起来，实现个人博客馆资源的最大化利用。另外，个人博客管理系统还可以节省人力、物力和时间。本系统满足了用户需要，方便其之间的相互交流。对数据库操作的性能需要进行优化，因为数据库过大将会影响运行速度[5]。

**1.5 开发的目标**

针对个人博客信息量大、信息处理复杂等问题，本文利用Node.js技术，开发适合大多数人使用的个人博客管理系统，建立了个人博客管理系统整体结构，完成了博客分类管理、个人博客操作管理、博客管理以及博客评论管理等功能模块的设计。

**第二章 系统开发环境**

针对个人博客管理系统信息量大灯问题，该章提出了采用Node.js技术标准的设计理念，介绍了MongoDB数据库、Node.js和Tomcat服务器的含义及特点等知识。

**2.1 MongoDB数据库**

MongoDB数据库毫无疑问是目前最主流的数据库之一，开源免费，已经被世界第二大软件开发和提供商甲骨文公司收购，获得了更多的商业支持，但是依然开源，免费的本色也没有改变。MongoDB最大的用武之地不仅仅是在服务器方面，在web应用方面也有出色的表现。它不单单是和php具有很强的通用性，对于Node.js，asp.net众多开发语言，能够将数据放到不同的表格中，是一款非常经典的关系型数据库，大大提高了安全性和灵活性[6]。

MongoDB数据库的安装包很小，但是这丝毫不影响写入、读取数据时速度快，同时还具有开放源码的优势。许多大中型公司的网站选择MongoDB数据库不但可以节省成本，在功能方面也能满足需求。因此本次毕业设计也选择了MongoDB作为数据储存。

MongoDB用于存放系统中的大量数据，MONGODB5.1.0支持HP-UX、Linux、Mac OS、NovellNetware、OpenBSD、OS/2 Wrap等多种操作系统；提供TCP/IP、ODBC和JDBC等很多数据库连接途径；能够对拥有很多条记录的大型数据库进行运算。当前由于MONGODB5.1.0在性能以及功能方面的优越性，已经成为众多中小型网站开发的首选，针对其开发的很多图形界面控制程序也不断出现，这将使得其开发和应用更加方便，因此具有更加广阔的应用前景。

**2.2 Node.js简介**

Node.js(Node.js Server Pages)是由Sun Micro systems公司倡导、许多公司参与一起建立的一种动态网页技术标准。Node.js技术有点类似ASP技术，它是在传统的网页HTML文件中插入Node.js程序段和Node.js标记，从而形成Node.js文件。用Node.js开发的Web应用是跨平台的，既能在Linux下运行，也能在其他操作系统上运行[7]。

Node.js与Node.js Servlet一样，在服务器端执行，通常返回给客户端的就是一个HTML文本，因此客户端只要有浏览器就能浏览。Node.js页面由HTML代码和嵌入其中的Node.js代码所组成。服务器在页面被客户端请求后对这些Node.js代码进行处理，然后将生成的HTML页面返回给客户端的浏览器。Node.js Servlet是Node.js技术的基础，而且大型的Web应用程序的开发需要Node.js Servlet和Node.js配合才能完成。Node.js具备了Node.js技术的简单易用，完全的面向对象，具有平台无关性且安全可靠，主要面向因特网的所有特点。Node.js具有跨平台、分离静态内容和动态内容、可重复使用组件、沿用了Node.js Servlet的所有功能、预编译等特点[8]。

Servlet是Node.js编写的服务器端程序，是由服务器端执行和调用的Node.js类。Servlet是采用Node.js技术来实现CGI功能的一种技术，能够像CGI一样动态地扩展Web服务器的功能，并采用请求——响应模式提供Web服务。Servlet和CGI一样都是运行在Web服务器上，用来生成Web页面。Servlet的主要功能在于交互式地浏览和修改数据，生成动态Web内容。

**2.3 Bootstrap**

Bootstrap 最初是由就职于 Twitter 的一个设计师和一个工程师创造的，现在，Bootstrap 已经成为了这个世界上最流行的前端开发框架和开源项目。

Bootstrap 最初是由 @mdo 和 @fat 于2010年中旬创造的。 在开源之前被称为 Twitter Blueprint。经过几个月的开发，Twitter 举办了首届 Hack Week 并将这个项目被公布出来，各个技能水平的开发者都可以在没有任何外部指导的情况下参与进来。在开源之前的近一年多的时间里，这个项目作为公司内部工具开发的样式指南，当然，现在它仍然扮演者同样的角色。

首个版本发布于 2011年8月19日（星期五），目前已经发布了超过20个版本，包括 v2 和 v3 两次重构。在 Bootstrap 2 版本，我们为整个框架添加了对响应式布局的支持，但是，这是作为一个可选的样式表而提供的。在前一版的基础上沃恩又发布了 Bootstrap 3 版本，我们再一次重构了整个框架，并将“移动设备优先”这一理念深刻的融入到整个框架中。

**2.4 Swig**

Swig 是node端的一个优秀简洁的模板引擎，类似Python模板引擎Jinja，目前不仅在node端较为通用，而且在浏览器端也可以很好地运行。

相对于其他js模板引擎有它的好处：   
1、与jade相比，更简单，jade是一个处女座的人才可以完全适应去写的，跟Python有的一拼；而且jade是一种面向对象的模板引擎，看起来并不是那么像是html了，但是关于性能的问题，尚未深入研究过。   
2、与ejs相比，ejs写起来比较麻烦，格式傻傻分不清楚，满眼的符号，尤其是他的{}，和jsp有的一拼：

**第三章 需求分析**

**3.1 可行性分析**

本系统为个人博客系统，显而易见，系统的核心对象是博客文章，系统需要大量存放博客的各种信息，而信息量不仅庞大还具有关联性。因此，数据库的使用很适用于该系统。数据库的使用环境以及配置如今都可以满足，使用起来也不是很难，不是一个技术难题。本系统的设计是以Node.js为开发平台，使用MyEclipse 8.5为配置环境。

**3.1.1技术可行性**

本系统是通过Node.js技术和MongoDB数据库技术，Node.js具有面向对象的优良特性，对于复杂程序的实现只需要通过更加简单逻辑就能实现。尤其是在安全性、可移植性以及多线程等方面的优势，使得Node.js语言的应用更加广泛。开发环境是MyEclipse，这些技术和开发工具，操作起来比较简单，没有什么难处。

**3.1.2经济可行性**

本系统主要采用Node.js和MongoDB等技术联合开发。他们有个共同点，就是免费。所以经济上永远不用担心软件版权费的纠纷。通过分析，个人博客系统在经济方面是可以执行的。

**3.1.3操作可行性**

本系统采用了精简的界面设计理念，就是为了让用户更为直观地感受本系统。让所有信息简单明了地展现在用户面前由此可见。所占用的资源量极小，普通的家用电脑无论在硬件上还是软件上都可以流畅地运行本系统，因此可以证明，本系统的合理运行是没有任何问题的，也就是说可行性良好。

**3.1.4法律可行性**

首先用到的资料都是合法的，而且本次所有的开发软件都是免费的，在开发的时候，所有的界面和设计流程也都是笔者自己构思的，不涉及到知识产权的问题，另外也有抄袭别人的系统的相关问题，所以在法律上是可行的。

**3.2 功能需求**

针对上面给出的Node.js技术准则，该章在对系统进行需求分析的基础上，对系统设计和数据库设计的内容进行详细的论述。

在开发系统的时候，要明确系统的开发需求，因此需求分析在系统开发的过程中具有至关重要的作用，后边所有的设计开发工作都需要基于需求分析进行。其主要任务是“目标系统要做些什么才能解决问题”。简言之，就是系统目标需要具有哪些功能。在这个至关重要的步骤里，程序员需要从用户那里搜集所有关于对系统需求的信息，以用来制作管理员和用户都普遍接受的系统说明书，通过系统说明书能够让人们理解系统是怎么操作的，这样能够减少操作中的错误。

根据功能将系统划分为如下及格个模块：

（1）用户管理模块，将用户的对象分为两类，管理员，用户，这一模块用来添加、删除、修改用户信息。

（2）博客分类管理模块，博客分类的管理是能够按条件查询博客信息，并能够对个人博客进行操作。

（3）对个人博客实施管理操作，比如说删除、添加信息等。

（4）博客文章管理模块，博客文章模块是本系统的核心部分，除了能够按文章名称模糊查询到博客的基本信息，还要实现对博文的增删改查操作。

（5）博客评论管理模块，博客评论管理在与用户可以对博客进行评价或者写出心得体会等等，一定程度上对其他的用户提供参考价值。

个人博客系统的使用者为用户和管理员。人们获取各种各样的信息，个人博客可作为其中一种十分重要的信息来源。

管理员实现对个人博客的管理、个人博客状态的管理等。整体来看，若是不用该系统，管理员可能没有那么多精力，更无法实施维护和运作。本系统可使个人博客的人力资源的管理更加的完善和人性化。

**3.3 性能需求**

数据精度要求：数据需要遵循严格的格式，否则系统将不会予以响应。

时间要求：运行环境方面的连接响应时间应小于等于10秒，对大于180秒没进行任何操作的用户其访问认证将失效，需要重新进行登入。

数据管理要求：即数据库系统管理，其包括：

注册信息表包含用户名，密码，住址，爱好，职业等基本信息。并且要求数据库的自动增长比例为2%。

日志记录：包含标题，时间，内容，作者等信息。

留言管理：包含回复人，内容，时间等信息。

控制：用户在进入博客首页面后，可以以三种身对个人博客站系统进行浏览——游客身份，普通用户身份（需要通过登录）和管理员身份（需要以特定的帐号来登录），在访问over后关闭所浏览的网页的同时也就退出了博客，

**第四章 系统设计**

**4.1概述**

本博客系统采用B/S结构，即Browser/Server或浏览器/服务器结构；建立在两种Web服务模式之上，本模型结构适用于Internet环境下。用户只需要连上Internet,便能够在任何时间、任何地点使用本系统，本网站工作的原理如4-1：

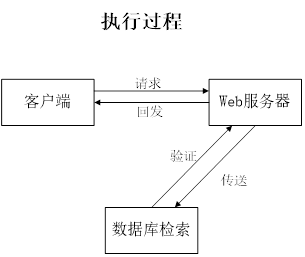


图4-1 系统工作原理图

**4.2系统结构**

**4.2.1技术选型：**

后台：Node.js（Express）

数据库：MongoDB

前端Js库：jQuery

前端UI框架：Bootstrap

前端模板引擎：Swig

**4.2.2功能：**

本个人博客站系统基于B/S架构，分为用户前台和管理者后台这两大模块。

前台的关键任务是：查看日志、查看相册、大家评论、音乐背景、用户登录、用户资料等功能模块。

后台的主要功能为：文章管理、留言管理等功能模块。

本系统的功能部分结构如图4-2所示：

个人博客

公告管理

评论管理

博客文章管理

系统设置

分类管理

公告添加

公告设置

留言板

点赞

发表博客文章

博客文章设置

文章分类

分类内容添加

分类内容设置

用户注册

信息设置

图4-2系统功能结构图

从上图可以看出本个人博客系统由前台和后台这两个功能模块组成。前台功能模块包括：

用户登录：用户通过自己所注册的账号和密码登录成功后便能进入属于自己的博客管理界面。

留言板：本博客系统具有留言板功能，游客和用户能均可以到留言板发表留言。

用户资料：对于登录之后的的用户来说，在这里能够修改自己的个人信息，对于么有登录的用户则只能浏览个人信息。

留言管理：博主可以在留言板以博主身份回复前台的留言信息。

相册管理：可以进行添加相册，修改相册，删除相册等操作。

图片上传：用户可以选择指定的相册，将上传图片至该相册中。

**4.3 数据库设计**

E-R图体现了不同主体之间的关系，代表了各个部分之间的功能和联系。设计合理的数据规划，可以提高效率，使数据存储的更加完善，保证了数据的完整和统一。根据对上述的分析，本文绘制了如下的E-R图，包含数据库项设计、数据表说明、数据表结构。

**4.3.1 E-R 图**

就此阶段而言，用户是能够参与到数据库系统的设计过程里的，还能对其进行平均价，进而确保数据库的设计能够与用户的实际需求相匹配。就概念模式的设计而言，最为普遍的设计方法属于E-R模型法[11]。结合以上对于数据库需求的分析，并在充分的结合的系统概念模式的特征以及开发方法的基础上，能够构建出的E-R模型图，此系统数据库E-R图如图3.6所示。

用户

操作

评论

相册

留言

好友

类型

文章

属于

包含

1:n

1:n

n:1

1:n

1:n

**4.3.2 数据库结构**

表4-1 添加用户表（users）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文字段名称 | 英文字段名称 | 类型 | 是否索引 | 备注 |
| 用户ID | \_id（主键） | ObjectID | 是 | NOT NULL |
| 用户名 | username | String | 否 |  |
| 密码 | password | String | 否 |  |
| 是否管理员 | isAdmin | Boolean | 否 |  |

表4-2 添加文章表（contents）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文字段名称 | 英文字段名称 | 类型 | 是否索引 | 备注 |
| 文章id | \_id(主键) | ObjectID | 是 | NOT NULL |
| 用户id | userid（外键） | ObjectID | 否 | NOT NULL |
| 类别id | category | ObjectID | 否 | NOT NULL |
| 发布时间 | addTime | Date | 否 |  |
| 标题 | title | String | 否 |  |
| 文章描述 | description | String | 否 |  |
| 正文 | content | String | 否 |  |
| 评论 | comments | Array | 否 |  |
| 访问量 | views | Number | 否 |  |

表4-3 添加分类表（categories）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文字段名称 | 英文字段名称 | 类型 | 是否索引 | 备注 |
| 类名ID | \_id（主键） | ObjectID | 是 | NOT NULL |
| 分类名称 | name | String | 否 |  |
| 分类排序 | sort | Number | 否 |  |

**第五章 系统实现**

**5.1用户登录模块的实现**

该模块主要是为了实现注册登录功能，主要是需要检测用户注册的用户名或者密码是否符合系统的设计要求。当用户注册或者登录成功，将跳转到主界面，当注册或者登录失败时，会及时给出提示信息。

相关代码：



操作与doRegister函数差不多，不一样的就是查询数据的时候，需要匹配账号和密码是否正确，登录成功之后都要写上session。

**5.2博客首页**

在个人博客的前台首页设计中，首先将管理员发布的最新文章展现给用户，然后提供文章的分类、显示博客的信息、显示最新公告及个人博客导航区功能等功能。个人博客前台首页的运行结果如图5.2所示。

本次设计着重点是NodeJS，并加之设计。博客主页虽然简单，但重要的是内容。图片的吸引力远远比文字大得多，所以尽量就避免了使用图片，实现最简单化。

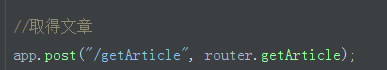


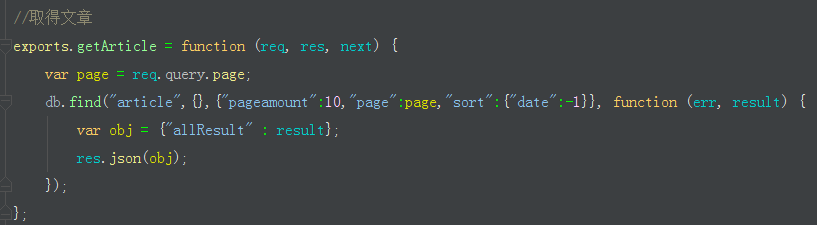
关键代码

前端显示的动态数据有一部分是Ajax获取，也有一部分是用模板渲染出来的。

**index.ejs首页**的动态数据都是从Ajax请求回来对上JS模板输出出来的。

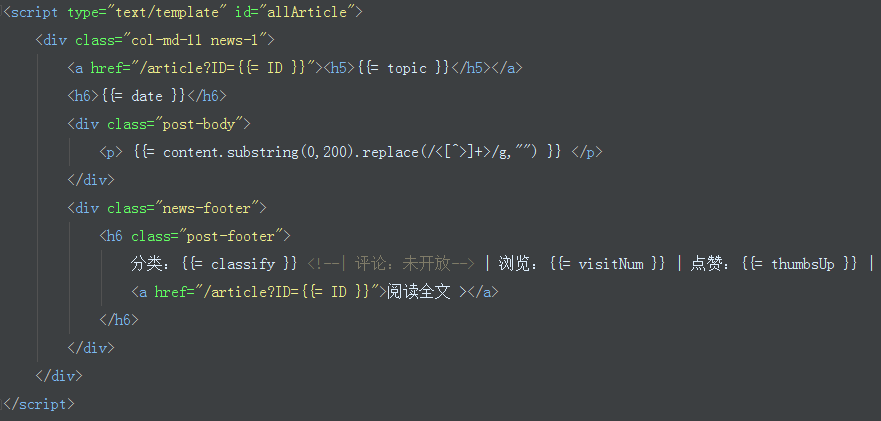
后台NodeJS对于POST请求回来之后的处理函数：





在这里看到每次请求都只是返回10条数据，这样可以大大优化查询的数据，然后用数据返回JSON格式。

Node.jsScript模板：



Ajax获取数据放到Node.jsScript模板里，由Node.jsScript渲染出来：

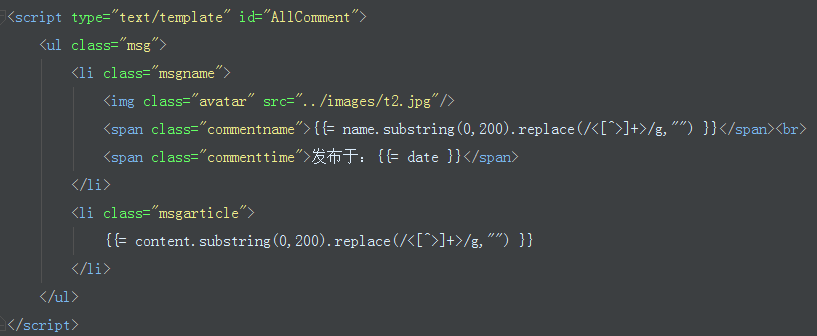


**5.3博客留言板**

在展现文章页面的末尾，提供接口供浏览者针对该文章发表评论，技术简单使用，就是提供发表评论标题和评论内容的文本框，让读者输入评论标题和内容，当读者单击[提交]按钮的时候，把评论标题和内容保存到数据库。



关键代码

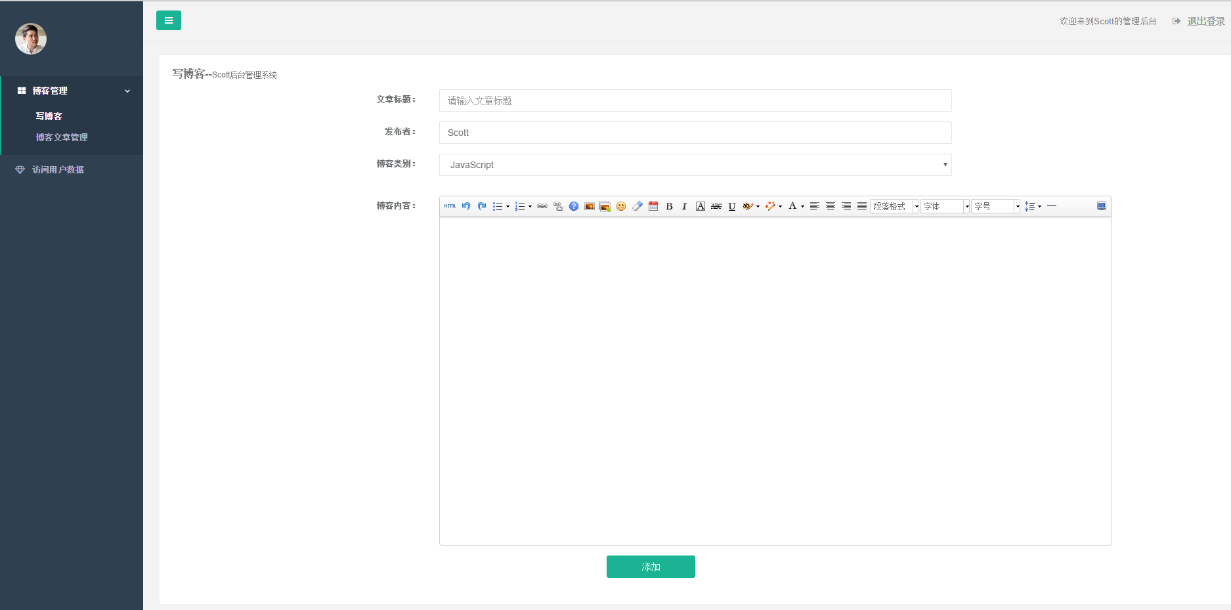


评论区页面的发表评论，就是Ajax提交数据，后台接收数据保存数据库。具体代码可看评论页面的模板：[comment.ejs](https://github.com/scottjeremy/EJblog/blob/master/views/comment.ejs)

在这里我们看到渲染数据的时候，首先{{}}里面第一个有个 “=” 的意思是以字符串的形式显示出来，不会执行HTML代码（包括脚本），加上replace是双重防止XSS攻击。

**5.4博客管理后台（写博客页面）**

在博客管理系统中，最重要的就是文章标记与发表功能了。此模块实现了博客的文章发表功能，包括文章标题，文章内容，发布日期，分类，以及标签等。文章发表成功后，会跳转到文章预览界面，也就是普通用户所浏览的页面。已发布的文章也可以通过此页面修改或者增减内容。



关键代码



首先访问该URL要判断是否有session，没有session就是代表没有登录，需要登录。服务器有session了，就渲染这个模板。

编写文章：将文章添加到数据库需要查询所有的文章，主要我还是想写个自增ID，想要文章的ID号。NodeJS里没有自增ID。只有随机的ID编号。然后存储的数据都是以JSON的格式，以语义化来命名，代码清楚明白。

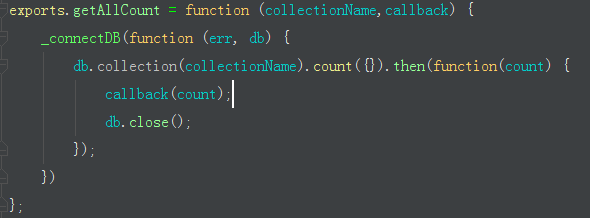
**5.5博客后台管理页面（文章管理）**

后台管理模块是博客用户特有的模块，此模块包含了文章管理，评论管理，以及标签等子模块的管理。主要是让博客用户更加方便的来管理自己博客的内容，通过对有歧义的文章或者评论进行删除或者屏蔽等处理，使得博客的内容更加符合博客管理者的思想。

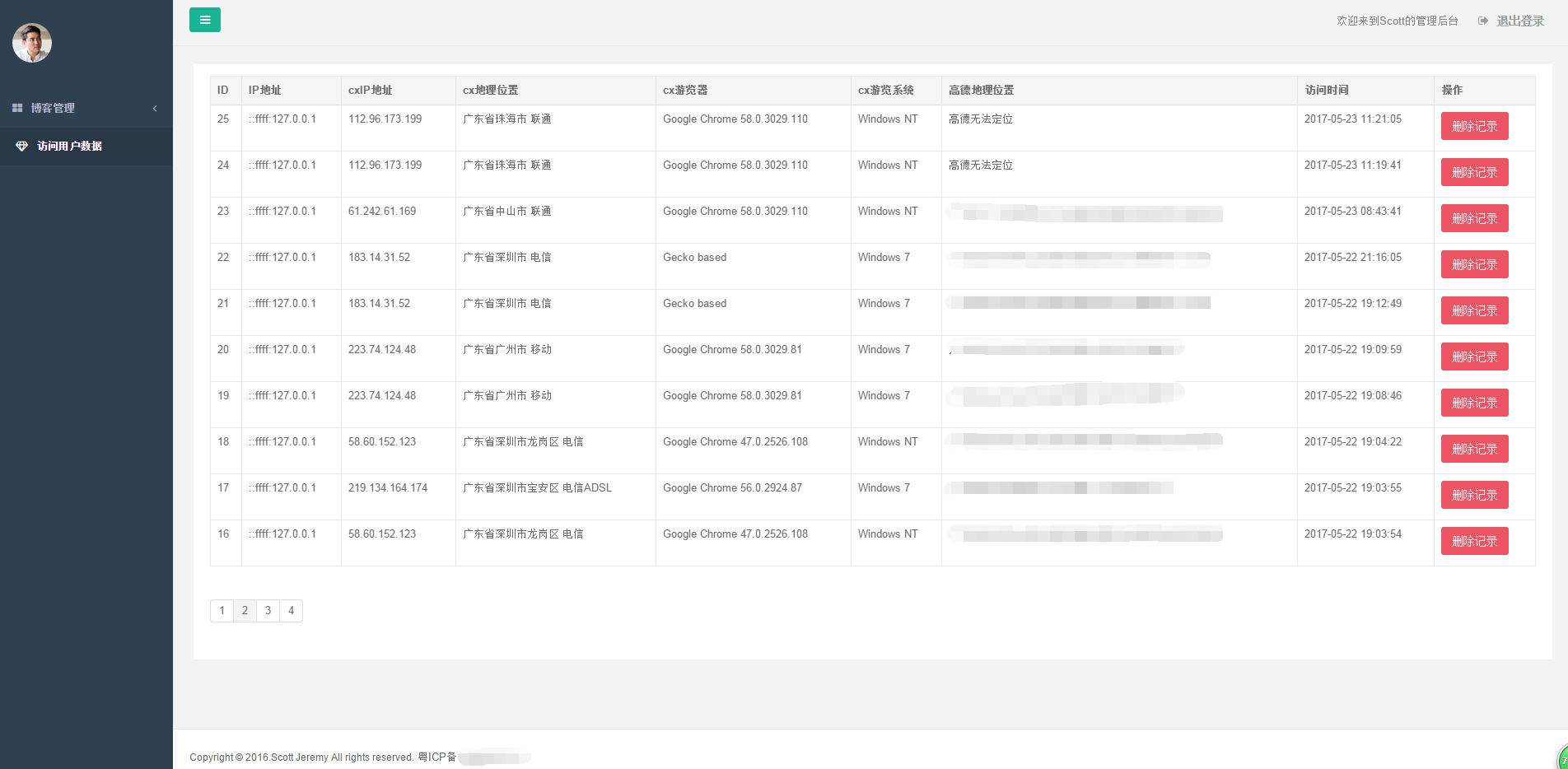


分页：

查询出来的数据肯定要分页。所以也有了这个函数。



**5.6博客后台管理页面（用户数据）**

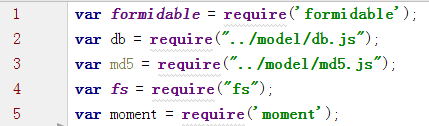


基本功能都在这里了，该作品纯碎是个人练手NodeJS所做出来的一个博客，如有建议，望告知。

## 5.7其他关键代码实现

**5.7.1引用文件**

router.js，包含了整个个人博客的精髓。在这里，所有的请求配置，返回的参数等等所做的业务逻辑都在这个文件里实现。



formidable是一个很出名的NPM包。它在node里作为表单内容接收做处理的一个中间件，功能很强大。

db.js是我在网上仿写的一个对数据库的统一操作的函数库。

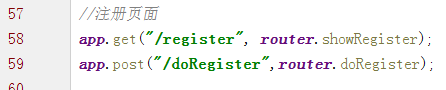
md5是用于数据加密的文件（现在反对用md5加密，因为可以被破解）推荐用哈希算法加密的中间件**Crypto**

fs是nodejs自带的一个文件管理，全名叫 fileSystem

moment是一个时间格式化的中间件

**5.7.2注册组件**

app是引用Express 然后引用Express里的get和post方法，由代码可以看出：第一个参数是路由设置 第二个参数就是请求处理函数。而下面代码就是引用router文件下的showRegister函数



exprots是ES6的新写法，可以在外面暴露这个函数，要引用这个函数前提是要引用这个文件

下面这段代码引用



 NodeJS都是回调，所有请求处理函数 都会接收三个参数：request（请求）,response（回应）,result（结果）

nodejs里有内置的render函数，可指定模板文件。而文件格式只能是ejs。

doRegister函数

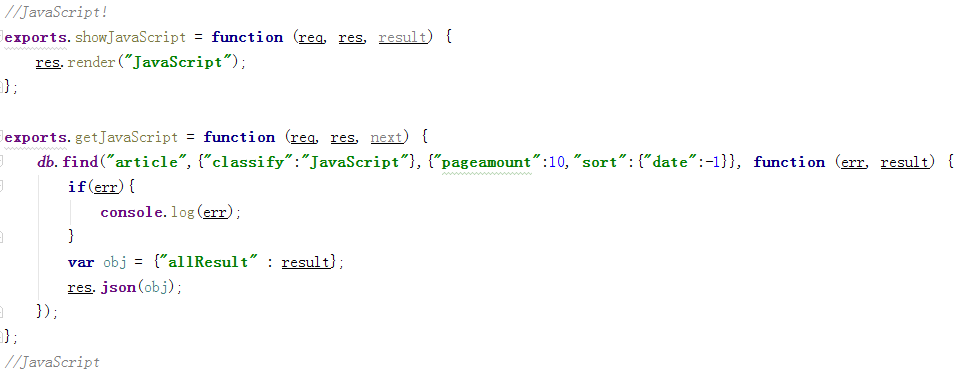
首先实例化formidable的IncomingForm，实例化对象变量名为form，引用实例化对象下的parse方法（接收两个参数：第一个，req，第二个，函数，所得到的表单信息都在fields里面，err参数就是错误信息，files就是文件信息，取到值之后就定义一个变量存储。密码加上双重MD5加密（然而并没有用，还是会被人破解。MD5不再安全），然后用db外置函数insertOne来向数据库新增一条数据，参数：（表名字,要存储的JSON,回调函数），而回调函数里又接收两个参数（是不是崩溃了？是就对了，因为我当初也崩溃了。）参数：(错误信息，结果)，如果错误则返回错误信息，成功就执行session设置并返回注册成功信息

**5.7.3获取用户地理位置组件**

用了查询网的API通过IP地址来判断访问者当前地理位置和客户端的信息。因查询网的定位不准确，用高德的游览器定位来做了这个功能。这里是在首页使用Ajax把数据发送到NodeJS接口来存储访问者的当前地理位置



**5.7.4分类组件**



在这个分类组件里，其余的都是一样，只是查询的时候多出了个条件。classify为Node.jsScript，就实现了文章分类的效果。

db.js文件主要是封装一些对数据库的连接操作，要不然都写在router.js上会造成非常大的代码冗余而且代码都一样。

对数据库操作，第一步当然是连接数据库：

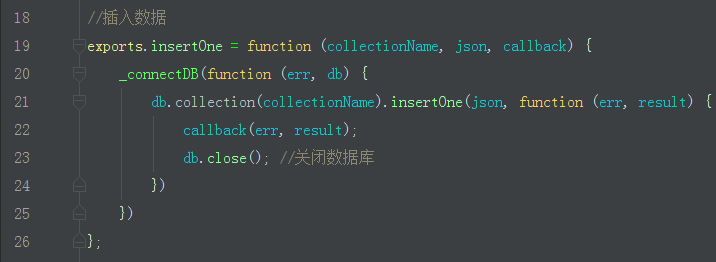


\_\_connectDB作为一个内部函数，方便被引用。

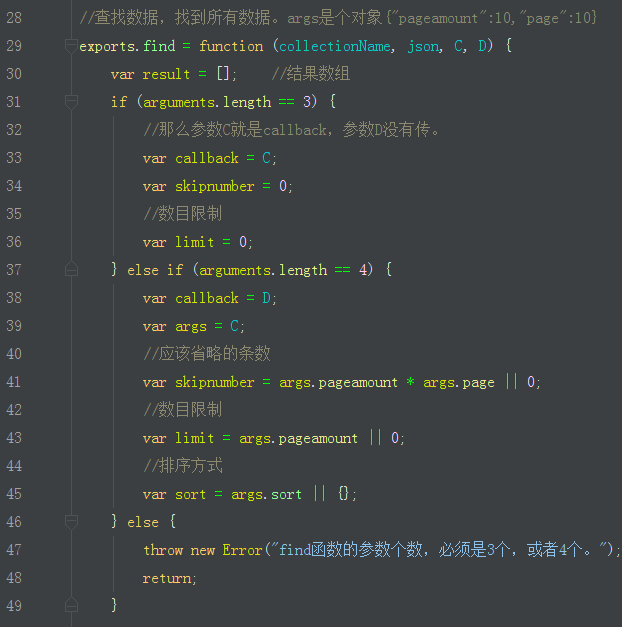
然后用MongoClient.connect所提供的方法连接数据库，connect接收两个参数（地址，回调函数）

**5.7.5增删改查基本操作**

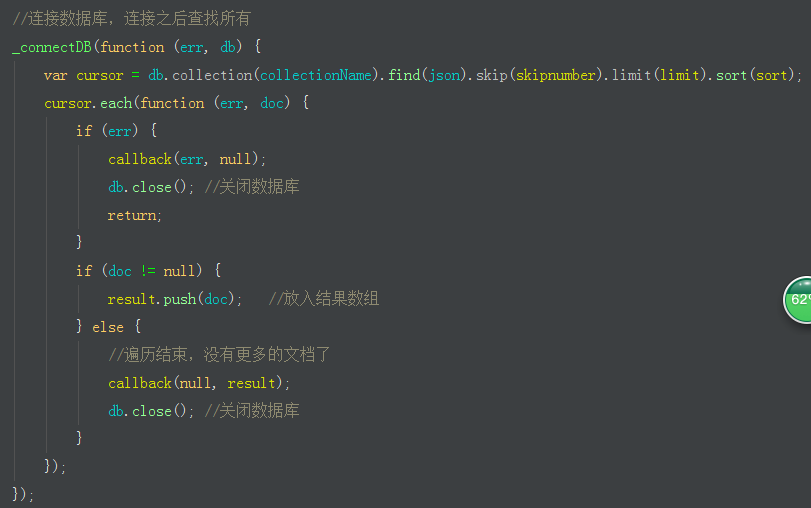
增：在数据库里插入数据，当然不能只限制一个，肯定要多条数据插入。所以insertOne函数接收三个参数（数据库名，要存储的JSON，回调函数）每次对数据库做操作都要连接一次而且做完操作之后一定要关闭数据库。因为每一条连接都会对数据库造成一定的压力，如果不释放这些空闲的压力，数据库的压力越来越会影响整体的性能。



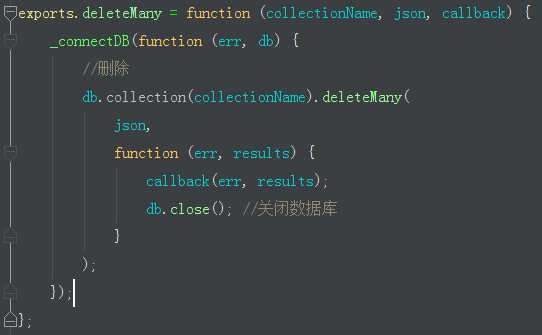
查：这里分两部分。第一部分（做一个判断，判断传来的参数有多少个）



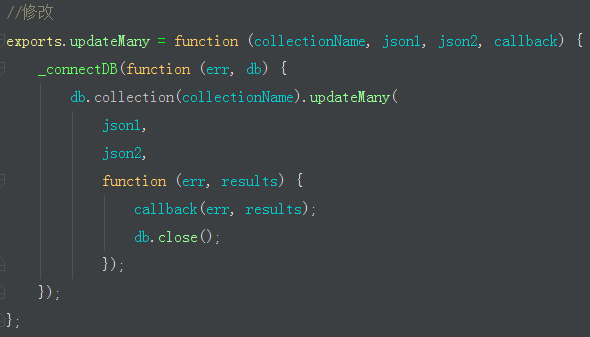
第二部分（查询数据库）：



删：从数据库中删除。接收参数（数据库名，要删除的JSON，回调函数）每次操作都需要一个回调函数来关闭数据库。



修：updateMany方法接收四个参数（数据库名，要修改的JSON，修改后的JSON，回调函数）最后还是关闭数据库。



**第六章 总结**

经过近三个月的努力，终于完成了这项程序设计及论文的撰写。设计成品能够符合我的设计初衷，即为特定年龄层用户以及博客初次使用者编写一款简单方便的个人博客系统。

本系统所实现的具体目标如下：

1、界面简洁、操作方便：使用完全控件式的页面布局，上用户在使用博客时更加简单方便；用户在使用博客的一些功能时只需按流程点击鼠标即可，不存在复杂的操作功能，实现博客易操作的目标；系统风格朴实简单，避免因花俏缭乱给人以不实之感。

2、即时可见：对博客进行的操作，包括发布、修改、删除等，一旦操作成功将立该在系统的相应位置出现，实现系统消息的提交与发表之间的零时差性。

3、功能完善：本系统包含一般博客系统所具备的三个模块：一般用户浏览、注册用户操作、管理员管理。不仅具备每个博客都有的写日志、上传照片、回复留言、修改信息等模块，甚至后台管理员可以对管理系统注册用户，对用户发布的日志、照片、视频进行审核，违反发布原则的可以立刻进行删除。

4、动态管理：能够动态的完成对数据库作息的存储、搜索、修改等操作，实现对系统的动态管理。

在查阅了大量的博客资料后，将书本上的理论知识同我制定的基本目标与功能想结合，设计出一套理想的实施方案。这样不但将以前所学过的知识进行了巩固，而且为今后从事软件开发工作做好准备。

本系统的界面采用简洁清爽的风格，系统功能让人一目了然，操作简单易上手。在系统功能实现方面主要使用Node.js+MongoDB 数据库等开发技术。在设计早期由于对一些知识有些欠缺，在老师的帮助下参考有关博客、解答相关疑问、纠正设计错误等，使我能够顺利的完成程序的设计。

经过这几个月的学习，我现在已经基本掌握了Node.js+MongoDB等网站开发技术，完成了从入门到运用的阶段跨越。除此之外，在程序设计过程中我还学会了做事要态度严谨、心思细腻，不然在过程中出现任何一个微小的错误都可能造成最后失败的结果。

由于时间和能力有所限，本系统必然不是完美的，程序和论文都存在不足之处。我今后将会更加努力学习软件开发方面的知识，争取将这个设计尽可能的改进完善。

**致 谢**

岁月如歌，光阴似箭，大学生活即将结束。经历了找工作的喧嚣与坎坷，我深深体会到了写作论文时的那份宁静与思考。回首两年多的求学历程，对那些引导我、帮忙我、激励我的人，我心中充满了感激。

首先要感谢导师李林，论文定题到写作定稿，倾注了李老师超多的心血。在大学期间，深深受益于吴老师的关心、爱护和谆谆教导。她作为班主任，点拨迷津，让人如沐春风;作为长辈，关怀备至，让人感念至深。能师从吴老师，我为自己感到庆幸。在此谨向老师表示我最诚挚的敬意和感谢!

在此要感谢我生活学习了两年多的母校——北京理工大学，母校给了我一个宽阔的学习平台，让我不断吸取新知，充实自己。

**参考文献**

1. 王越. 基于nodejs的微博系统的设计与实现[D]. 电子科技大学.
2. 彭娜. 基于Node.js博客系统的设计与实现[D]. 大连理工大学, 2013.
3. 姚立. IBM云计算平台下NodeJS应用支持环境的设计与实现[D]. 哈尔滨工业大学.
4. 尹延宁, 刘太君, 叶焱. 基于Node.js,Sencha Touch和iBeacon信息推送系统的设计与实现[J]. 火力与指挥控制, 2016, 41(8):151-154.
5. 单振华, 王舒憬, 陈凯. 基于NodeJS的智能家居语音控制系统服务器端设计与实现[J]. 工业控制计算机, 2016, 29(4):66-67.
6. 刘派. 基于Node.js的热计量监控预警系统的设计与实现[D]. 北京邮电大学, 2014.
7. 赵正凯, 杨烁, 李欧. 基于Node.js技术的综合测评系统的设计与实现[J]. 电脑知识与技术, 2014(19).
8. 王倩[1] , 许成鹏[2] , 惠小强[3] . 基于node．js的学习管理系统的设计与实现[J]. 电子设计工程, 2018.
9. 邹竞莹. Node.js博客系统的设计与实现[D]. 2016.
10. 王金龙. 基于Node.js的图片分享应用的设计与实现[D]. 2015.
11. 叶树健, 伍仕宝. 基于Node.js平台的移动报修系统的设计与实现[J]. 工业控制计算机, 2017(06):113-115.
12. 黄志峰, 许尚文, 肖慧. 基于Web与Node.js的医用磁共振设备动态管理系统的设计与实现[J]. 中国医疗设备, 2018, 33(10):31-34.
13. 鄢化鹏, 聂云峰, 汪涛. 基于Node.js的违章查询系统设计与实现[J]. 软件导刊, 2018, v.17；No.189(07):136-138+142.
14. 王伶俐, 张传国. 基于NodeJS+Express框架的轻应用定制平台的设计与实现[J]. 计算机科学, 2017(S2):606-609.
15. 牟雅琳, 丁浩哲, 郭茂林. 基于Node.js的大学生自助学习答疑系统的设计与实现[J]. 电脑知识与技术, 2018, v.14(21):113-115.
16. 罗文, 刘星, 兰全祥. 基于Node.js的创新团队管理系统的设计与实现[J]. 赤峰学院学报(自然科学版), 2018, 34(10):89-91.
17. 向宇. 基于Node.js的智能锁管理系统的设计与实现[D].
18. 柳志强, 陕粉丽. 基于NodeJS的聊天系统的设计与实现[J]. 电脑知识与技术, 2017(13).
19. 鄢化鹏. 基于Node.js与Android的违章查询系统设计与实现[D].
20. 赵亮, 赵绪辉. 基于NodeJS的跨平台聊天应用的设计与实现[J]. 电脑迷, 2017(1):28-29.
21. 戴鹏程. 基于Android与Node.js的车友会系统设计与实现[D].
22. 王壮. 基于Nodejs框架的企业人事管理系统的设计与实现[D].
23. 王亚强. 基于Node.js的网购交流系统设计与实现  [J]. 电脑编程技巧与维护, 2016(24):27-30,共4页.
24. 仇晶, 黄岩, 柴瑜晗. 基于Node.js中间层Web开发的研究与实现——以微信图书借阅平台为例[J]. 河北工业科技, 2017(2).