 **JIANGXINORMALUNIVERSITY**

本科生毕业设计（论文）

**基于PHP的个人博客的设计与实现**

**Based on the design and implementation of a personal blog of PHP**

**学院名称：计算机信息工程学院**

**专业名称： 网络工程**

**姓 名： 谢志强**

**学 号： 201308093051**

**指导教师： 周新宇**

**完成日期：**

## 声明

本人郑重声明：

本人的毕业论文（设计）是在指导教师指导下独立进行研究取得的成果。其中引用的地方都加了标注和致谢，在法律规定允许的情况下，不包含任何其他个人或集体已经撰写完成或已经发表的以某种方式公开过的研究成果，不包含为了获得教育机构的证书或学位而作的材料。任何对本研究有贡献的个人和集体都在文中作了明确的说明并且表达了十分的谢意。本人完全意识到本声明的法律责任由本人承担。

本人的毕业论文（设计）成果在江西师范大学读书期间，在指导教师精心的指导下取得的，成果归江西师范大学计算机信息工程学院所有。  
 特此声明。

声明人（毕业设计（论文）作者）学号：  
声明人（毕业设计（论文）作者）签名：  
  
签名日期：　　　　年　　月　　日

## 摘要

随着科技发展，网络在人们的工作和生活中扮演了越来越重要的角色。各类网络交流平台得到了广泛关注和应用，其中博客作为一种高粘性的网络交流平台拥有大量的使用者。通过博客，人们可以自由地表达出自己的观点，和网友一起交流和讨论。并且，通过博客也可以结交到很多兴趣相同的朋友，具有很高的实际使用价值。

本文主要介绍一个个人博客系统的开发与设计，从系统开发背景、系统需求分析、系统环境搭建、系统代码实现等这几个方面来介绍整个系统的开发过程以及一些技术细节。本系统采用PHP、MySql、JavaScript、HTML、CSS等常用Web开发技术，使用了基于Node.js的webpack前端打包工具自动部署前端应用，尝试运用流行的前端技术框架Vue.js来构建前端页面。系统以扁平化风格为基调，简洁大方，符合现代化的设计风格。

**关键词**：Web；JavaScript；Git；Node；个人博客；

## Abstract

With the development of science and technology, the informationization slowly into the center of the society, people gradually through a variety of network communication platform for information exchange. Blogs are a great place to shine as an emerging online platform. People like to manage the blog, because they can in their own blogs to express their views and win the respect of others, on the one hand is to filter and accumulate knowledge, on the one hand, also can to blog marketing for media themselves, at the same time can make a lot of like-minded bloggers. Blogs can help people with difficulties, and have a high level of shared spirit and value.

This paper mainly introduces the development and design of a personal blog system, from the system development background, system requirement analysis, system environment construction, system code to realize these a few respects the development process of the whole system are introduced, and some technical details. This system uses common web development techniques such as PHP, MySql, JavaScript, HTML, and CSS. Using the node.js webpack front-end packaging tool to automatically deploy the front-end application; try using the popular front-end technology framework Vue. Js to build the front-end page. The system is the basic tone with flat style, concise and easy, accord with modern design style.

**Key words:** Web; JavaScript; Git; Node; Personal blog;

## 目录

[声明 I](#_Toc482278016)

[摘要 II](#_Toc482278017)

[Abstract III](#_Toc482278018)

[目录 IV](#_Toc482278019)

[第1章 绪论 1](#_Toc482278020)

[1.1 系统开发背景 1](#_Toc482278021)

[1.2 系统开发目的 1](#_Toc482278022)

[第2章 需求分析 2](#_Toc482278023)

[2.1 系统目标 2](#_Toc482278024)

[2.2 系统需求 2](#_Toc482278025)

[2.2.1 经济需求 2](#_Toc482278026)

[2.2.2 技术需求 3](#_Toc482278027)

[2.2.3 功能需求 3](#_Toc482278028)

[第3章 开发平台介绍与环境的搭建 6](#_Toc482278029)

[3.1 开发平台 6](#_Toc482278030)

[3.2 WAMP架构 6](#_Toc482278031)

[3.2.1 WampServer 介绍 6](#_Toc482278032)

[3.3 MVC 7](#_Toc482278033)

[3.4 MVVM 8](#_Toc482278034)

[3.5 前后端分离 9](#_Toc482278035)

[3.6 开发环境与工具 9](#_Toc482278036)

[3.6.1 前端Node环境 9](#_Toc482278037)

[3.6.2 后端Wamp环境 10](#_Toc482278038)

[3.6.3 版本管理工具Git 10](#_Toc482278039)

[第4章 系统实现详述 12](#_Toc482278040)

[4.1 模块设计 12](#_Toc482278041)

[4.1.1 系统整体模块 12](#_Toc482278042)

[4.1.2 数据库设计 12](#_Toc482278043)

[4.2 前台 15](#_Toc482278044)

[4.2.1 UI设计 15](#_Toc482278045)

[4.2.2 整体实现 16](#_Toc482278046)

[4.2.3 细节优化 18](#_Toc482278047)

[4.3 后台 19](#_Toc482278048)

[4.3.1 后台登录 19](#_Toc482278049)

[4.3.2 文章管理模块 20](#_Toc482278050)

[4.3.3 博客汇总模块 22](#_Toc482278051)

[4.3.4 个人信息模块 23](#_Toc482278052)

[4.4 项目总结 27](#_Toc482278053)

[参考文献 28](#_Toc482278054)

[致谢 29](#_Toc482278055)

## 第1章 绪论

### 1.1 系统开发背景

随着科技发展，二十一世纪网络化，信息化慢慢转变成社会的中心，人们逐渐通过各种的网络交流平台来进行信息交流。博客作为新兴的网络交流平台大放光彩。博客，它的正式名是网络日记，是一个自主管理、不定时间的发布新文章的系统。博客的内容是自由的，可以是对时政的看法或者是课题的深入研究，也可以是生活感悟、旅游、视频、音乐、学习笔记等。博客因为它自由开放和信息共享的文化特征，获得了很多的使用者。人们开始热衷于管理自己的博客，因为他们可以在博客上表达出自己的观点而赢得别人的尊敬，一方面是对所学知识技能进行过滤和积累，一方面也是可以把博客当做一种媒介来营销自己，同时也可以结交到很多兴趣相同的朋友。虽然博客具有较强的个人性，但它可以帮助到他人，也是具有很高的共享精神和使用价值。博客的出现成为人们分享和获取信息这种社会需求的解决办法。

### 1.2 系统开发目的

系统的设计初衷在于提供一个信息沟通、交流与分享的平台，系统的设计目的有以下几个方面：

1. 作为自己的日记，可以在记录总结自己的经历的同时锻炼自己的表达写作能力。

2. 开发系统时尝试用一些新的技术，学习新技术的同时也是个人知识技能的总结与应用。

3. 作为自己想法的集结地，在博客中尽情地表达出自己的想法，表现出自己的创新意识，激发自己的潜能。

4. 可以提高自己的影响力，博客汇聚了许多杰出的人，一起讨论某个领域的研究，好的博客可以提高自己影响力，赢得别人的赞许和尊敬。

5. 作为自己结交朋友的一种方式，通过博客系统这个平台，可以结交到很多的博客好友，同时也提供一个给他人获取信息的平台，促进与他人信息的交流。

## 第2章 需求分析

需求分析是本系统开发中非常重要的一个步骤，同时也是决定系统能否开发完成的一步。需求分析往往决定了系统的开发方向、技术选择以及是否符合预期，做好需求分析对于系统开发来说极为重要。

### 2.1 系统目标

博主登录到该系统，可以方便快捷的创建博客分类，发布博客，并且对创建的数据信息有良好的增加，删除，修改，查询的操作体验，可以方便的维护系统其他数据，如维护个人信息、友情链接、公告等的信息。系统还应提供给游客良好的界面，更加人性化的设计，让游客可以快速的查找到感兴趣的博客信息，对游客具备交互的友好性。

### 2.2 系统需求

#### 2.2.1 经济需求

下面对系统开发中的经济情况进行分析，采用的开发工具和所需硬件需求如下。

硬件需求：

(1) PC 机：

内存：DDR3 4G

硬盘：500G

网络连接：有线网卡 / 无线网卡

其他要求：标配

(1) 操作系统：Windows系列操作系统

(2) 数据库平台： MySQL数据库系统

(3) 开发平台：Node + wamp

(4) 浏览器：chrome、360极速浏览器

(5) web 服务器：Apache

(6) 其他开发工具：wampSever、WebStorm、PhpStorm、Navicat、Git软件等。

综上所述，本系统开发需要的的软件和硬件投入成本较低。

#### 2.2.2 技术需求

对系统开发而言，技术可行性分析是很重要的一个环节。对本系统开发采用的技术的分析如下。

1. HTML、CSS、JavaScript 等基本网页开发技术。

熟练地运用html、css、javascript等技术快速的构建出基本的系统页面。对技术的使用有模块化的思想，能够对开发中的一些共通点进行抽象、分离、封装。

2. AJAX 技术

系统都是采用ajax技术与服务器进行数据的传输。Ajax就是创建一个XmlHttpRequest对象（现代浏览器），浏览器通过XmlHttpRequest对象向服务器发起异步请求，浏览器从服务器获得数据后，再用javascript来操作要修改的DOM节点，更新需要更改的页面[1]。

3. HTTP 数据请求

系统的数据的传输是采用http协议的，如http的get和post请求。http作为常用的网络数据协议，需要掌握它的工作原理和实际使用中的细节。

4. SQL 数据库

Mysql数据库是开源的轻量级的数据库，对于数据的存取也是可以满足本系统的需求的，数据库的操作语言的话是采用sql语言，后台框架对于数据库操作语言也有封装，主要运用是表的查询和表的写入等。

5. PHP 动态语言开发技术

采用php作为后台语言的开发技术，并且用thinkphp这一基于MVC框架模式的php框架来快速的搭建系统后台。

6. Node 相关技术

采用基于Node.js平台下的webpack前端打包工具，来作为前端自动化部署的工具，使用webpack可以快速的对静态资源进行模块化打包，同时对一些新的规范（es6等）和一些拓展语言（sass、less等）进行转化打包。

#### 2.2.3 功能需求

功能需求是指系统开发完成后一定要实现的功能，从而满足用户的实际的业务需求和体验。系统提出的主要功能如下：

前端：

首页界面：展示所有分类下的博客，并且按时间倒序展示。

博客分类页：展示某种分类下所有的博客，并且按时间倒序展示。

博主信息：展示博主基本信息，后台可以配置展示信息。

友情链接：展示最近添加的五条链接，按时间倒序展示。

系统公告：展示最新添加的系统公告。

后端

登录界面：提供用户登录后台的界面。

博客分类管理页：管理所有的博客分类信息，可以对博客分类信息进行增删改查操作。

博客内容管理页：管理所有的博客内容信息，可以对博客内容信息进行增删改查操作。

友情链接管理：管理系统友情链接信息，可以对友情链接进行增删改查操作。

系统公告管理：管理系统公告信息，可以对系统公告进行增删改查操作。

系统设置：管理系统运维的基本信息。

个人信息管理：记录博主最后登录时间、总计登录次数和最近登录日志，包括登录时间的记录、登录IP的记录、和是否登录成功的记录。还有对博主信息的查看修改、密码的认证修改。

系统功能需求图如图2.1所示

图2.1 系统功能需求

## 第3章 开发平台介绍与环境的搭建

### 3.1 开发平台

Microsoft Windows，作为操作简便、提供了良好的图形界面的操作系统，已经拥有了广大的使用者，随着windows XP的终结，微软已经不再维护windows XP系统，win7虽然不如win8和win10的界面炫酷，但凭借自身的稳定与快捷，仍然挽回了很多使用者，现在win7、win8还有win10已经成为广大电脑使用者的首选。而作为开发平台来讲，windows系统开发上确实不如linux或max等操作系统，毕竟很多技术都是在linux上开发而来，但是由于window系统的简便、性价比高，同时很多技术也都尽可能的去兼容windows系统，所以还是有广大的开发者支持windows系统的，本次个人博客系统的开发是在64位Windows 8.1系统下完成。但对于开发应用程序，还是推荐用linux或mac系统来进行开发。

### 3.2 WAMP架构

WAMP架构指的是Windows下的Apache、Mysql、PHP（取首字母命名），这些软件都是开源的、免费的、常用于搭建动态网站的开源软件[4]。 Apache（Apache HTTP Server）作为使用广泛的web服务器，用于解析系统网页语言，如html，php；Mysql（关系型数据库管理系统）是快速的，操作简便的数据库，适用于中小型企业系统数据库的首选，当然一些大企业的某些业务也是可以使用mysql来满足的，它主要用于系统的数据的存储和查询，mysql在使用者数量排行上也是一直处于领先地位的；PHP（Hypertext Preprocessor）作为后端动态编程语言，比起java，php在使用上更加的灵活，性能上也不输与java，可以使我们的网页动态化，将数据库中的数据按照我们的要求展示在网页上，让我们可以更加方便的管理我们的系统。这套架构广泛应用于中小型企业的网站搭建上，一些大公司则是采用其他应用架构，像Linux + Nginx + Oracle +Java这样的架构来搭建网络应用。

#### 3.2.1 WampServer 介绍

wampserver作为一个开源的、完全免费的、集成apache 、php 、mysql的软件包，对于网站开发的初学者而言，无疑是绝佳的一个选择，你可以不用麻烦地去安装一个一个的软件，处理软件安装和软件之间兼容性的问题。对于初学者来说，安装那些软件所需的时间和遇到的困难有可能会使开发者丧失对编程的兴趣，wampserver给我们带来了福音，wampserver提供了良好的安装的图形界面，你只需要跟着它的安装向导一路安装，最终可以方便快捷的开始自己的编程之路。当然它也提供了很多其他拓展的功能，你也可以通过图形界面去增加自己想要的功能模块，后续甚至可以自己添加一些额外的功能。你可以很轻松地在其官方网站下载到最新的版本。接一个下来你只要根据图形界面的提示安装就可以在windows平台上搭建一个apache、php、mysql 的集成环境。

### 3.3 MVC

MVC（Model-View -Controller）包括了三个模块，分别是模型、视图以及控制器。它的主要作用是将实际代码进行分离、抽象、封装。比如实际开发中的业务逻辑可以抽象为一个模块，数据处理一个模块，视图又是一个模块，模块与模块之间形成一个单向数据流。我们访问一个网页，根据路由我们可以访问到控制器，根据控制器调用数据模型来存取数据，在发送给视图来渲染数据，此时用户若在视图上进行操作，控制器又能够接收到用户操作，再次进行数据存取，视图渲染的一系列的操作。这样形成一个完整的单向数据流，可以使我们在开发中很方便的去调试我们的代码，可以让我们更加清楚数据的流向，可以快捷的定位出出错的地方[9]。Model（模型）用来处理系统的数据。通常表现为与数据库打交道，比如系统与数据库的增删改查操作，这些都是在模型中完成的。View（视图）是在界面上展示数据的，就是用户最直观的看到的系统的界面。Controller（控制器）是用来处理用户交互逻辑，控制器逻辑基本上是和具体的业务逻辑相同，用来响应用户操作，并向调用模型处理数据，在将数据整合到视图。Mvc操作流程如图3.1所示。

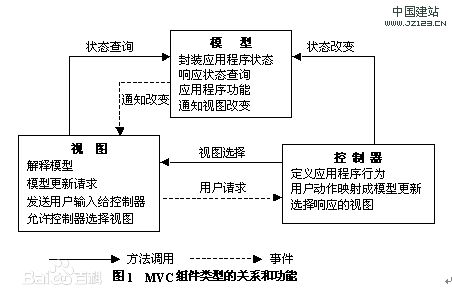


图3.1 MVC 流程示意图

### 3.4 MVVM

MVVM(Model View ViewModel) 是现在一种比较流行的替代前端mvc的方案，和mvc或mvp类似，mvvm架构中的model代表着的是系统的数据层，比如系统与数据库的增删改查操作；而view就是用户最直观的看到的系统的界面；最重要的是viewmodel，它将视图和数据模型绑定在了一起，相当于我们如果更改了数据，那么相应的视图也会发生相应的改变，同样我们如果更改了视图，那么数据也会进行相应的变化。有了mvvm框架，作为一名前端工程师，是可以感受到无尽的快感的，因为我们终于可以从复杂的dom操作中彻底地解救出来，它可以让前端工程师更专注于自己的业务逻辑[5]。如本博客中vue.js便是典型的mvvm框架。具体流程如图3.2所示。

BfyAFfF

图3.2 ViewModel 流程示意图

图中我们可以看到的是viewmodel流程示意图，它将视图（view）和数据模型（model）进行了绑定，现在无论是视图的变化，还是数据模型的变化，都可以相互之间通知并进行改变。

### 3.5 前后端分离

网站分为前端和后端。对于用户而言，前端指的是普通用户在浏览器中所看到的网站的界面，后端是指网站管理员为管理网站数据的后台界面。对于开发者而言，前端往往包括很多技术，如html、css、javascript（jquery、vue、react）等等，后端则是php、mysql之类。web研发模式也经过了多次演变，从后端为主的 mvc 时代到以前端为主的 mvvm 时代，逐渐的使前端工作专注在浏览器端，后端工作专注在服务端。前后端分工之后，让我们的系统开发可以同步进行，当然前端可能需要进行一些接口数据的模拟，比如可以建一个json文件或者其他手段都行的，后端也可以专心地处理实际的业务逻辑，给前端输出约定好的ajax接口。由之前后端渲染整个页面返回浏览器显示到前端负责整个页面后端负责提供ajax接口，使前后端分工明确，同时极大的程度上减轻了服务器的压力。

### 3.6 开发环境与工具

#### 3.6.1 前端Node环境

Node.js就是基于 Chrome V8 引擎的 JavaScript 运行环境。注意Node.js并不是一门语言，只是一个JavaScript 的运行环境[6]。随着前端工程量的加大，传统的开发模式已经不能满足广大前端开发者的需求，前端模块化乃大势所趋，从而孕育出一系列优秀的模块化js库，如符合amd规范的require.js和cmd规范的sea.js。因为javascript语言是弱结构性的，所以通过cmd和amd的模块定义方式来对代码进行模块化管理，这样使得js代码更易开发和维护；除此之外，nodejs对模块的规范定义以及包（npm）这个概念的引入，使得js模块化管理再也不是一件困难的事。随着js模块化的流行，前端工程师们在开发时往往将一个模块作为一个单独的js文件来管理，这时候网站上线时，为了使浏览器能够正确的解析一些新的规范（es6等）和一些拓展语言（sass、less等）同时也能减少网络请求的三次握手的数量，需要将各个模块进行转化打包成一个文件，这时候一系列基于node.js模块打包工具如雨后竹笋现世，如grunt、gulp、webpack、yarn。node可以到[官网](https://nodejs.org/en/)下载安装即可。

npm在安装node时已经安装好了。你可以输入以下命令来查看安装的版本，如图3.3所示。

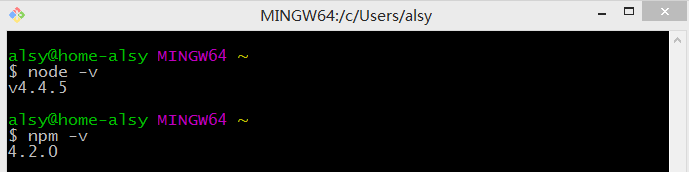


图3.3 node及npm版本查看

前端开发工具推荐采用webstorm、sublime等代码编写工具。

#### 3.6.2 后端Wamp环境

后端的wamp架构，可以一个个的安装apache、mysql、php这些网站开发的软件，当然这是比较费时费力的，有时还会遇到无法预知的问题，这是我们可以安装wampsever这一集成的软件，然后快速的开始我们的编码。安装完成后在浏览器（推荐chrome浏览器）地址栏输入localhost访问即可。后端采用基于mvc模式的thinkphp框架。推荐使用phpstrom来编写代码。

#### 3.6.3 版本管理工具Git

Git可以高速有效的处理或大或小项目的版本管理，是一个版本控制系统。在没有git之前我们开发人员的版本管理是十分繁琐的，我们为了能够保留每次的项目代码更改记录，会给项目进行编号之类的操作，然而随着项目的不断增大，修改记录的不断增加，维护这些版本号变得异常的困难，你会发现你已经陷入了版本号管理的地狱，无奈又无助。Git的出现宛如黑暗中的一道亮光，让我们看到的希望，使用git后，我们只需简单的命令就可以方便的进行版本的控制与维护，我们可以在Git[官网](https://git-scm.com/downloads)下载到最新的稳定版本的git，接下来只需要根据安装向导安装即可，如图3.4 所示。

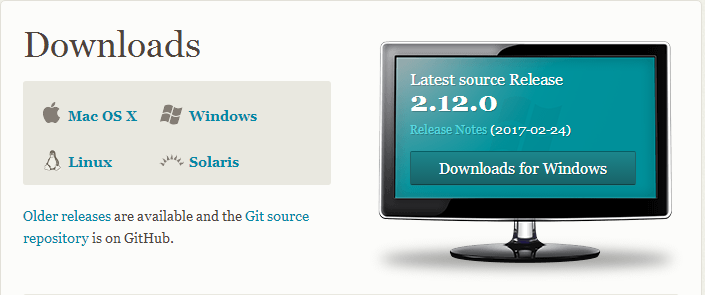


图3.4 Git官网下载页面

安装完成后，命令行输入以下命令查看git版本，看到打印出了Git的版本信息，意味着你已经安装成功了，接下来你可以很方便的使用Git来管理项目版本，如图3.5所示。

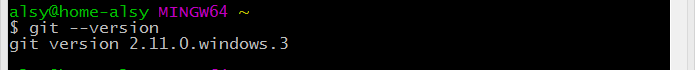


图3.5 Git版本查看

## 第4章 系统实现详述

### 4.1 模块设计

#### 4.1.1 系统整体模块

系统前台模块主要包括博客分类、博客内容、友情链接、系统公告、博主信息五个模块，对所有游客都展示。对应的后台有对五个模块的管理和博客汇总以及系统运维相关配置和系统信息的维护。如图4.1所示。

图4.1 系统整体模块示意图

#### 4.1.2 数据库设计

由五个基本模块对应到数据表的设计，如下：

1. blog\_article\_category（博客分类表）如表4.1所示。

表4.1 博客分类表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 功能描述 |
| id | int | 10 | 主键 |
| pid | int | 10 | 父级分类id |
| name | varchar | 255 | 分类名称 |
| rank | int | 3 | 权重 |
| index | int | 1 | 首页是否展示 |
| status | int | 10 | 状态 |
| remark | varchar | 20 | 备注 |

1. blog\_article\_detail（博客内容表）如表4.2所示。

表4.2 博客内容表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 功能描述 |
| id | int | 10 | 主键 |
| title | varchar | 255 | 博客标题 |
| category\_id | int | 10 | 分类id |
| user\_id | int | 10 | 博主id |
| content | longtext | 0 | 博客内容 |
| hits | int | 10 | 博客点击量 |
| file\_id | int | 10 | 文件id |
| is\_publish | tinyint | 2 | 是否发布 |
| create\_time | int | 10 | 创建时间 |
| update\_time | int | 10 | 更新时间 |
| status | int | 10 | 状态 |

1. blog\_broadcast（系统公告表）如表4.3所示。

表4.3 系统公告表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 功能描述 |
| id | int | 10 | 主键 |
| content | varchar | 255 | 公告内容 |
| create\_time | int | 10 | 创建时间 |
| status | int | 10 | 状态 |

1. blog\_friend\_link（系统友情链接表）如表4.4所示。

表4.4 系统友情链接表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 功能描述 |
| id | int | 10 | 主键 |
| name | char | 50 | 链接名称 |
| url | char | 100 | 具体链接 |
| create\_time | int | 10 | 创建时间 |
| status | tinyint | 1 | 状态 |

1. blog\_user（用户表）如表4.5所示。

表4.5 用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 功能描述 |
| id | int | 10 | 主键 |
| name | char | 20 | 博主姓名 |
| roles | char | 20 | 博主角色 |
| user\_no | char | 15 | 账号 |
| password | char | 150 | 密码 |
| sex | char | 1 | 性别 |
| phone | char | 60 | 电话 |
| hometown | varchar | 10 | 家乡地址 |
| github | varchar | 255 | Github |
| qq | char | 10 | QQ |
| email | char | 30 | 邮箱 |
| status | int | 10 | 状态 |
| session\_id | varchar | 255 | session\_id |
| login\_last\_time | int | 10 | 最后登录时间 |
| login\_times | int | 10 | 登录次数 |

网站角色权限、操作日志、网站信息、上传文件等表如下：

1. blog\_frame\_access
2. blog\_frame\_action
3. blog\_frame\_alert
4. blog\_frame\_file
5. blog\_frame\_log
6. blog\_frame\_role
7. blog\_frame\_session
8. blog\_frame\_setting

### 4.2 前台

#### 4.2.1 UI设计

首页、分类页设计[2]：主要是向游客展示博客分类、文章简短信息、公告、博主信息、友情链接等信息。点击某个分类则进入到该分类下，显示出该分类下的所有文章。其中导航条分类顺序以及友情链接顺序可以由后台配置，最新文章以及阅读排行榜都是自动读取数据库里5条最新记录和5条阅读量最高的记录，如图4.2所示。

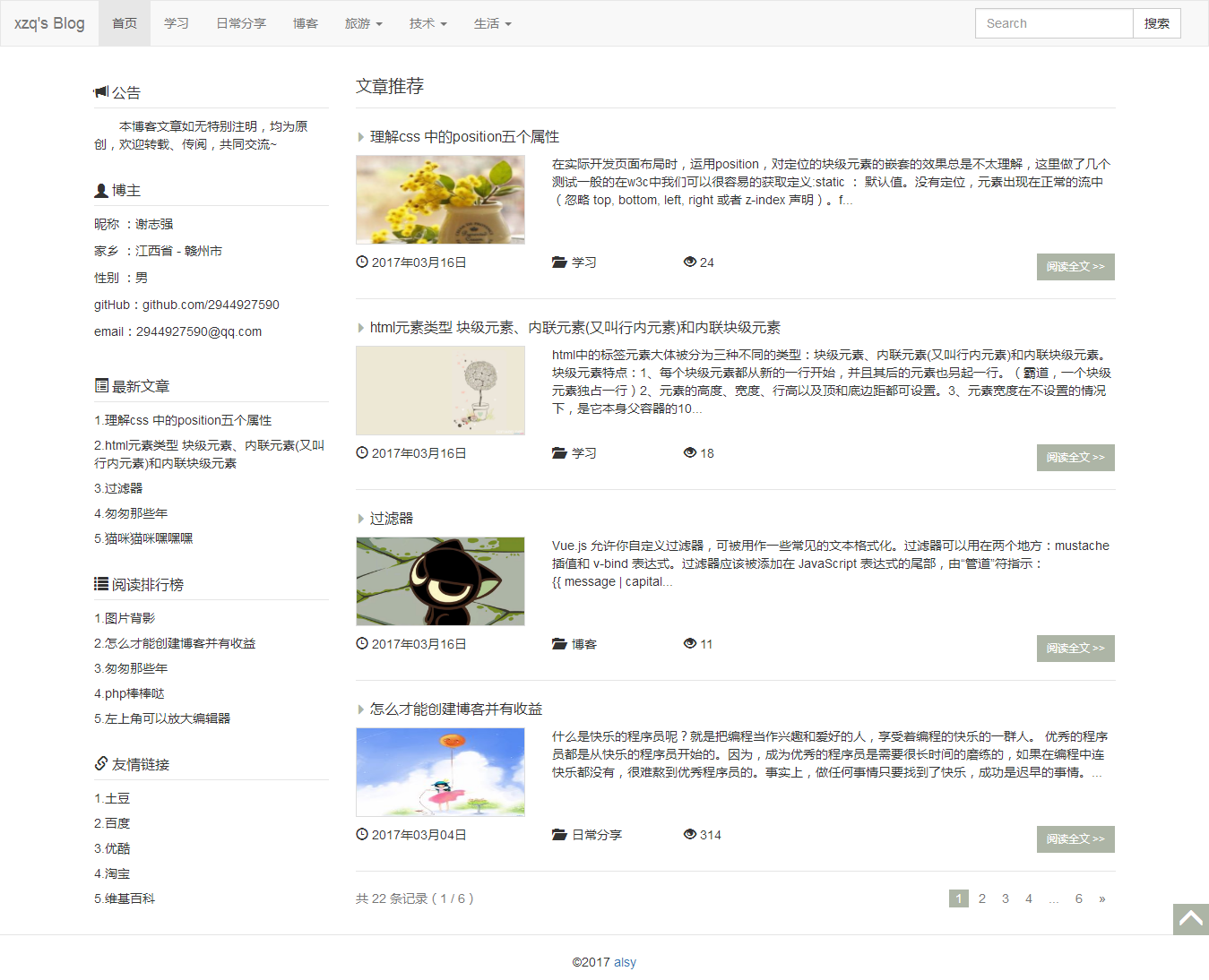


图4.2 首页及分类页设计

博客详情：主要向游客展示文章具体内容，文章内容由后台ueditor编辑器编辑，支持多种内容排版。如图4.3所示。



图4.3 详情页设计

#### 4.2.2 整体实现

页面采用bootstrap作为整体的样式基调。采用vue-cli 官方命令行工具。vue-cli 是vue.js提供一个强大的官方命令行工具，可以用来快速的搭建或大或小的单页应用。该命令行工具提供给你傻瓜式的构建工具配置，并且带给你现代化的前端开发流程。命令行安装如图4.4所示。

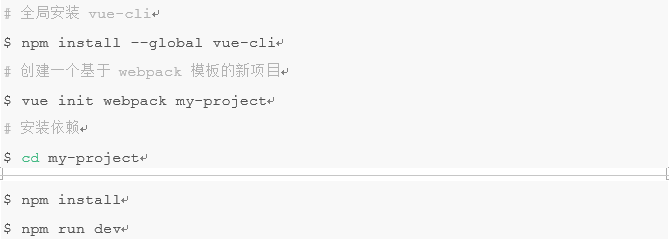


图4.4 vue-cli安装

可以看到系统根目录有个package.json文件，里面记录了系统基本信息，例如系统名称、系统版本、系统描述、系统作者等信息，同时记载了系统运行所必需依赖包。如图4.5所示，是本次个人博客系统中的package.json文件：

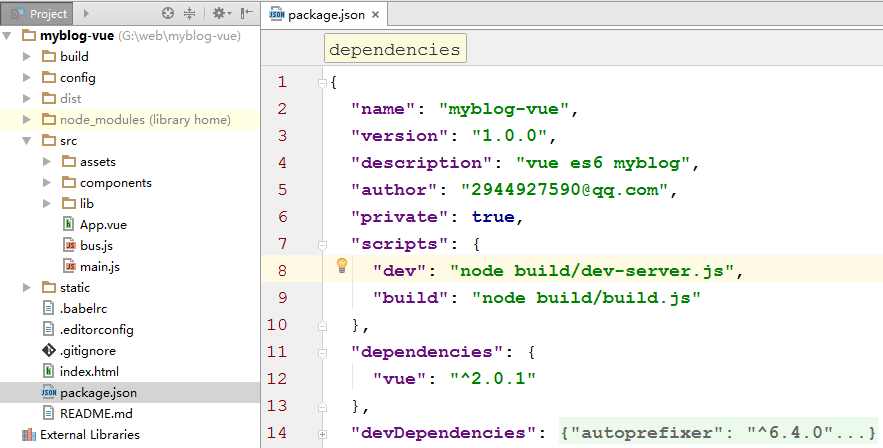


图4.5 系统package.json文件

命令行运行npm run dev（实际上运行的是后面的node脚本），即可运行系统，如图4.6所示。

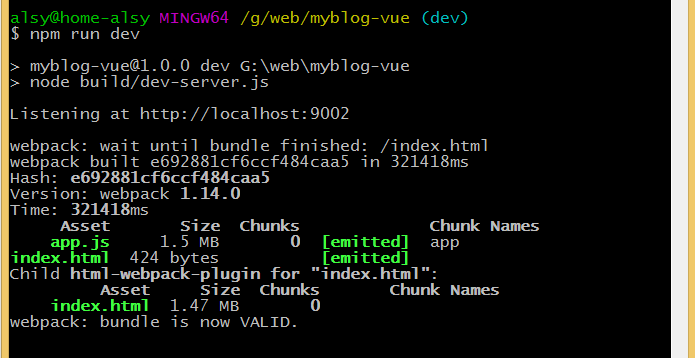


图4.6 命令行启动系统

浏览器中输入本地测试地址即可访问。

开发系统中将各个页面拆分成一个个的组件，再以搭积木的方式搭出一个个页面，每个组件之间耦合度极低，每个组件只关注自身的状态，组件可以复用，极大增强系统的健壮性，降低了后期维护的成本。系统采用vue-router 路由组件处理系统的路由，使得系统可以方便的根据路由来响应不同的组件。路由的配置如图4.7所示。

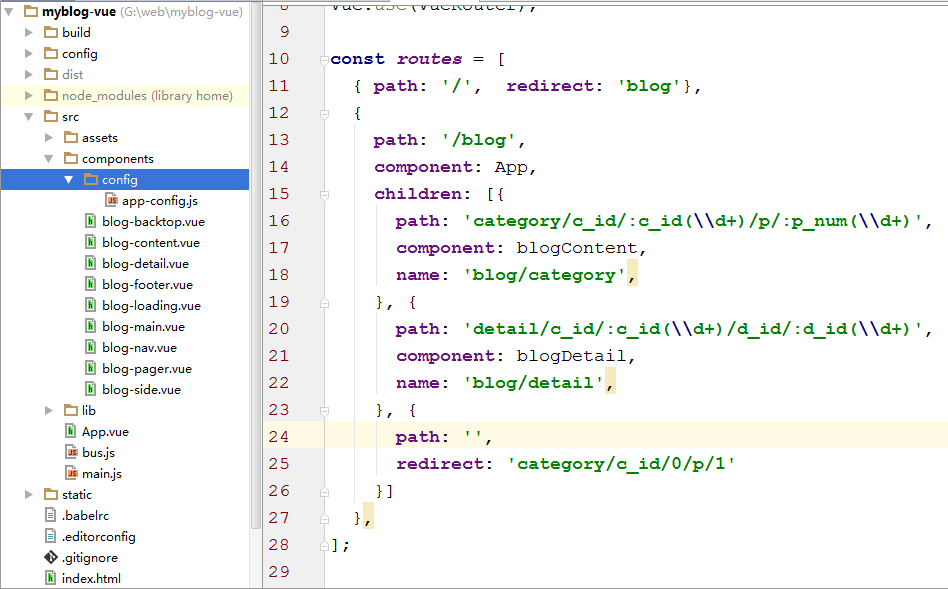


图4.7 路由配置

系统中所有数据都是由后台提供的ajax接口返回的json数据，同时对于一些网站常用而又不常变的数据，采用了localstorage浏览器本地存储数据的技术，极大的减少了网络请求，加快页面加载速度[3]。

#### 4.2.3 细节优化

1. 网页加载时提供loading动画，使得用户在操作时不会那么突兀。
2. 当页面滚动条离窗体顶部距离大于200像素时显示回到顶部的图标供用户快速回到顶部，同时做了防抖动处理，极大提高性能[10]。

核心代码如图4.8所示：

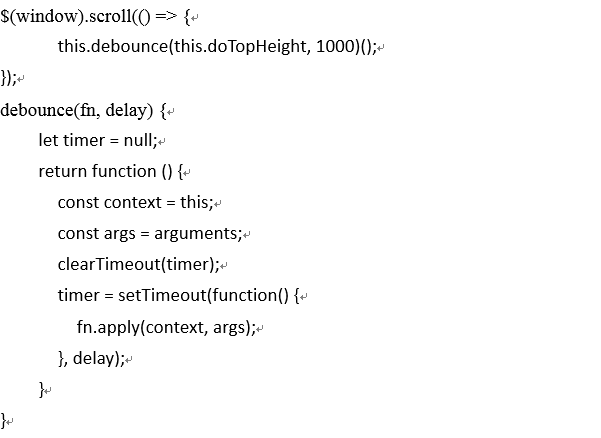


图4.8 防抖动处理

1. 点击页面加载后自动回到顶部，方便用户从头开始阅读。
2. 网页加上一些小图标，保持对用户友好的交互和视觉体验。

### 4.3 后台

#### 4.3.1 后台登录

输入后台网址，经过入口文件index.php，指引到System/HyStart/login 这个路由上，服务器渲染页面返回给页面显示[8]。这里的登录操作是以ajax方式向后台发起请求的，不会造成页面的刷新。账号、密码正确后进入后台首页。登录页如图4.9所示。



图4.9 系统后台登录页

#### 4.3.2 文章管理模块

在文章内容页面你可以查看文章的标题、所属类别、发布人、文章点击量、文章附件下载、发布时间、文章是否发布、文章状态。可以根据文章标题来进行模糊查询，类别以及是否发布来进行查询。可以把数据导出成excel表格，也可以进行打印操作。如图4.10所示。

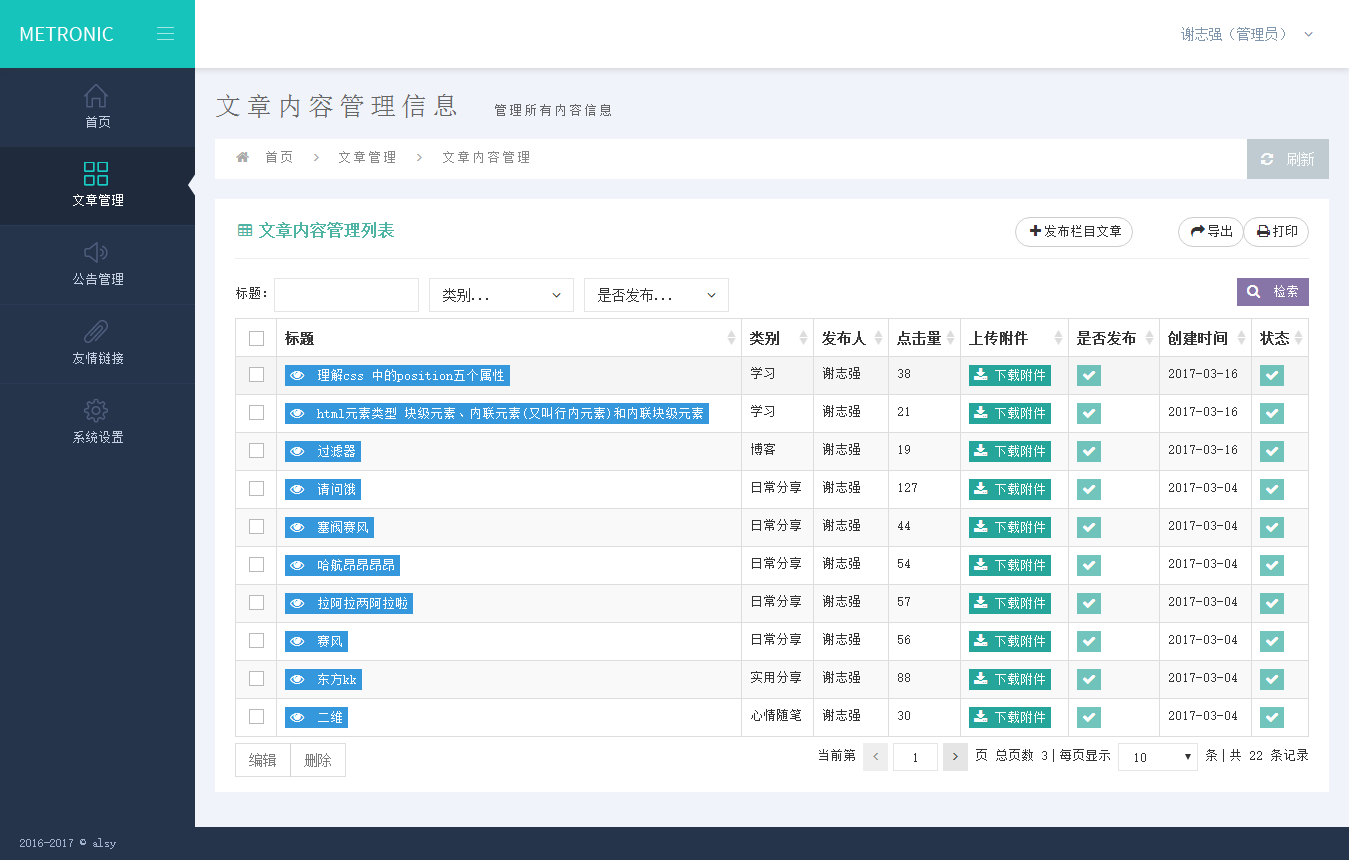


图4.10 文章内容页面

点击发布文章按钮，弹窗显示新增文章的页面，在这个页面填写文章的标题（必填2-100个长度的字符）、类别、内容、点击量（非必填必须是数字）等字段来快速新建一篇文章，其中必须符合字段的验证规则才可以提交成功。如图4.11所示。

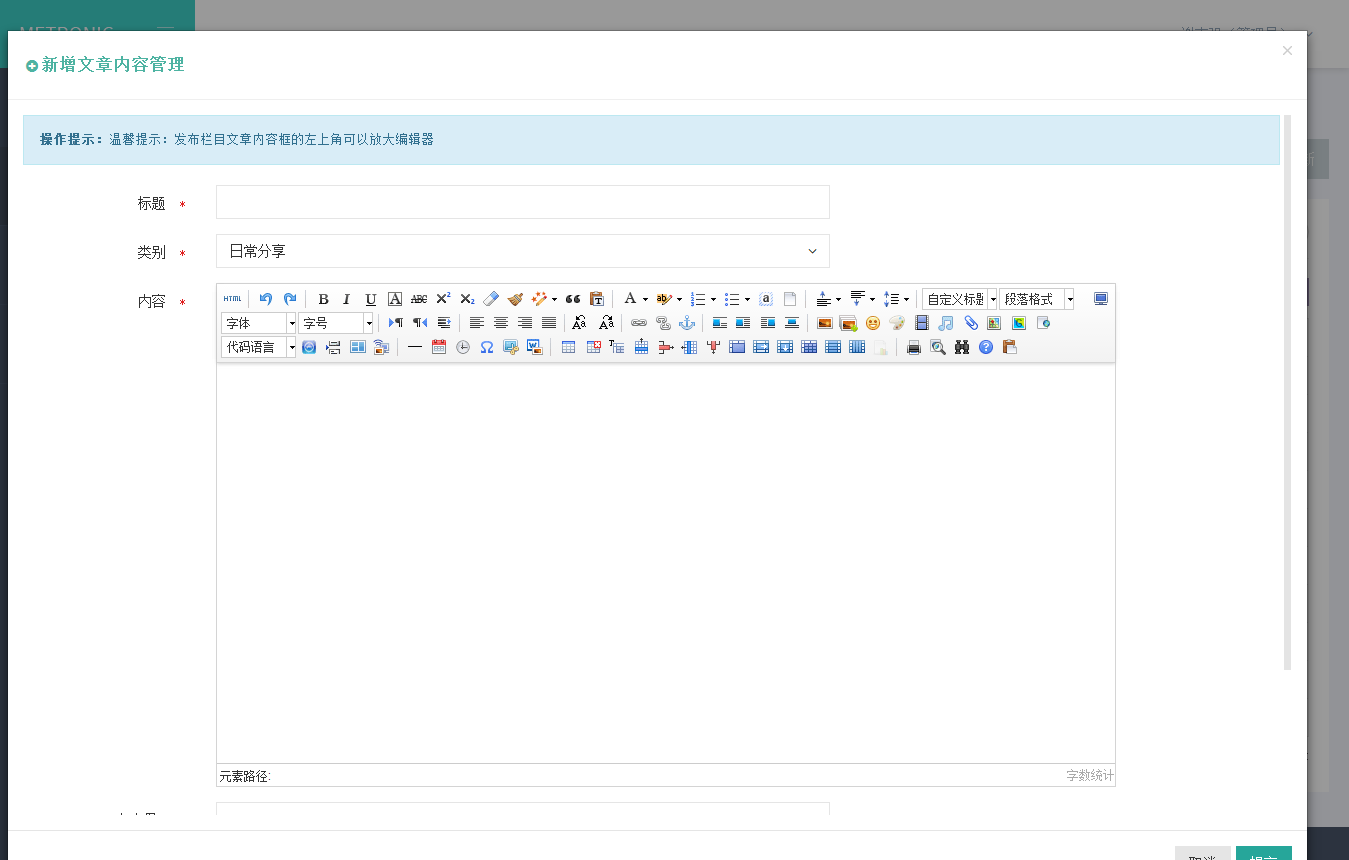


图4.11 新增文章页面

点击编辑按钮，弹窗显示文章内容编辑页面，在这个页面编辑文章的标题、类别、内容等字段来快速修改一篇文章的内容。同样的必须满足创建时的规则，如图4.12所示。



图4.12 文章内容编辑页面

在类似的这个页面，你可以进行搜索、新增、编辑、删除、导出、打印等操作。而且这些操作都是以ajax的方式操作数据，用户体验上更加快捷，人性化。而对于模块的配置，核心代码如图4.13所示。



图4.13 模块配置的核心代码

注：其他几个类似的模块都是这样处理，在此不做累赘描述。

#### 4.3.3 博客汇总模块

在系统的博客汇总模块中，可以非常直观的了解到一年中每个月每个分类下的博客数量，可以方便的了解到用户博客记录的整体情况。如图4.14所示。

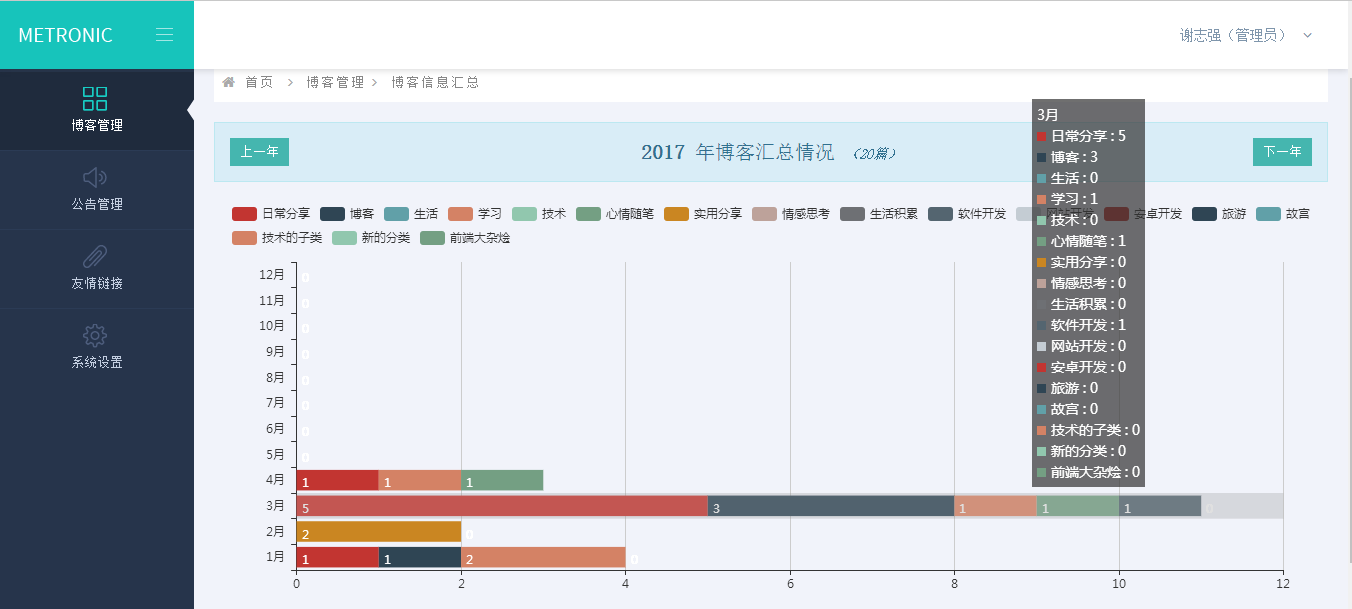


图4.14 博客信息汇总页

在这个模块，可以方便的查看每年每个月每个分类的博客数量，默认是当前的年份，同时也可以切换年份来查看。

#### 4.3.4 个人信息模块

1. 个人信息·概况信息页

在这个页面展示了博主最后登录信息、累积登录次数以及近期登录的日志，包括登录时间、登录IP、是否成功等信息。如图4.15所示。



图4.15 个人信息·概况信息页

核心代码如图4.16所示。

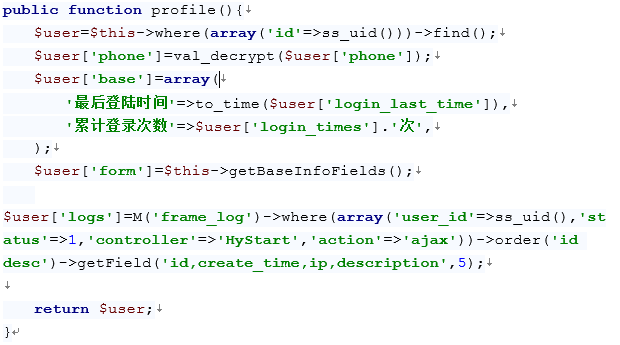


图4.16 个人信息·概况信息页核心代码

1. 个人信息页·概况信息·基础信息页

在这个页面展示了博主的基础信息，包括账号（必填的3-11个字符）、姓名（必填的2-30个字符）、电话（非必填的7-11个字符）、QQ号（非必填的5-10个字符）、电子邮箱（必填的）、家乡（非必填的0-30个字符）和GitHub（非必填的0-100个字符）账号。博主可以方便的对其中任意的信息进行修改。如图4.17所示。

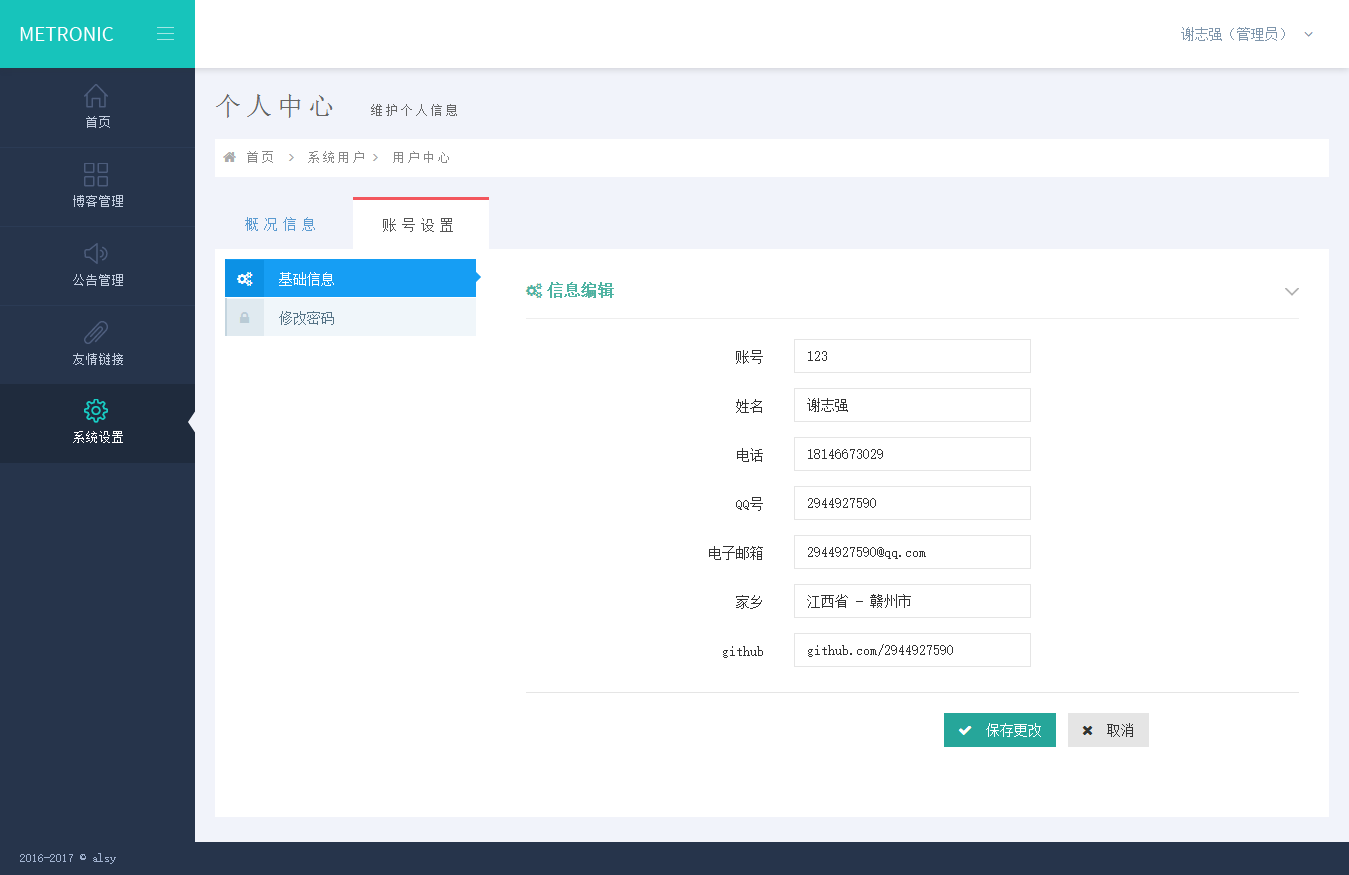


图4.17 个人信息页·概况信息·基础信息页

核心代码如图4.18所示。



图4.18 个人信息页·概况信息·基础信息页核心代码

1. 个人信息页·账号设置·修改密码

在这个页面博主可以对登录密码进行修改，系统要求输入旧密码进行核对，并且输入两次相同密码才能对密码修改成功，三个字段都是必填的。如图4.19所示。



图4.19 个人信息页·账号设置·修改密码

核心代码如图4.20所示。



图4.20 个人信息页·账号设置·修改密码核心代码

### 4.4 项目总结

这次系统设计对我来说获益匪浅，实现系统时采用了一些时下比较流行的工具和新的技术，比如基于Node的webpack构建工具和Vue.js的尝试，同时也把之前学的理论知识付诸于实践，自己对理论知识也有了反思和新的理解。在构建系统时也遇到过很多难点和技术上的生疏点，通过自己的思考和参考他人的经验逐一攻克，锻炼了自身学习能力和解决问题的能力，对于自身方法论的形成起到重要的影响。在不断完善系统的过程中，自己深刻的体会到要把一件作品做好，做精细是一件多么不容易的事情，这需要我们有匠人精神，有一种不断完善自己作品、坚持不懈、追求完美的精神。同时也锻炼了自己的勇气，对于新技术尝试的决心。体会到敢想就要敢做、要做就要做好的勇气和精神。

## 参考文献

[1] Douglas Crockford 编 赵泽欣译 JavaScript语言精粹(修订版) 电子工业出版社，2012.09.01

[2] 前沿科技 编 精通CSS+DIV网页样式与布局 人民邮电出版社2007.08.01

[3] (美)，David Flanagan 著 淘宝前端团队 译 JavaScript权威指南(第6版) 机械工业出版社2012.04.01

[4] 徐安令. 基于WAMP架构的博客管理系统分析与设计[J]. 电脑编程技巧与维护,2015.

[5] 陈涛. MVVM设计模式及其应用研究[J]. 计算机与数字工程,2014.

[6] 陈瑶. 基于Node.js高并发web系统的研究与应用[D].电子科技大学,2014.

[7] 李海峰. MVC模式架构的应用研究[J]. 自动化与仪器仪表,2013.

[8] 主福洋,郭坤. 基于PHP技术的网站建设[J]. 软件工程师,2013.

[9] 李容. 基于MVC模式的Web应用研究[J]. 软件导刊,2010.

[10] (美)，Nicholas C. Zakas 著 李松峰 ，曹力 译 JavaScript高级程序设计(第3版) 人民邮电出版社 2012.03

## 致谢

第一篇毕业论文的编写，过程中我的心情是十分复杂的。从一开始的选题、收集资料、到后来的程序编写、Bug调试、性能优化、细节优化，一开始选题的纠结，新技术尝试的困难让我一次次的想放弃这个题目，但是随着自己的坚持努力，终于完成了这一篇设计，现在回想起来，也是别有一番滋味啊。

在这里我想感谢我的母校也就是江西师范大学。在母校我度过了愉快的四年大学时光，在四年时光中我结交一群有激情有热血的朋友，同时得到了很多的成长，收获了很多。感谢我的导师周新宇老师，他为人热情和蔼，治学严谨细心。他的无私帮助，在我苦恼的时候给我不断的加油鼓励，在我困惑的时候为我解答问题，整个论文的定稿过程中有老师很大的心血。同时也要感谢我的大学辅导员和我的所有大学老师们，他们让我学到了很多的专业技能，让我可以了解到世界的精彩纷呈，让我学会如何的做人做事。感谢四年里陪伴我的同学、朋友们，他们是我人生中宝贵的财富，有他们我在奋斗的路上就不会孤独。

最后我要感谢培养我长大的父母，谢谢您们！最后的最后，再次对关心和帮助我的老师以及同学们表示衷心地感谢！