

**NANCHANG UNIVERSITY**



JavaWeb设计大作业报告

题 目： 个人博客

学 院： 软件学院

专 业： 软件工程

班 级： 软件工程（卓越）153班

完成人数： 1人

成 员： 贺胜利 8000115296

起讫日期： 2018/6/1~2018/6/24

任课老师： 周兴斌 职称： 副教授

完成时间： 2018/6/24

目录

[一、概述（包括项目背景、编写目的、软件定义、开发环境等内容） 3](#_Toc517641691)

[二、需求分析（问题陈述、需完成的功能。） 3](#_Toc517641692)

[三、系统设计（可用UML建模） 4](#_Toc517641693)

[四、程序原代码及其说明 6](#_Toc517641694)

[五、总结 35](#_Toc517641695)

[六、参考文献 43](#_Toc517641696)

### 一、概述（包括项目背景、编写目的、软件定义、开发环境等内容）

##### 1、项目背景

个人博客是当今网络的热点。个人博客技术的出现使得每个人都可以在零成本和零维护的情况下创建自己的网络媒体。博客网站形成的网络结构促成了博客文化的不同于以前的社区，博客技术创造了一个“博客”。

##### 2、编写目的

个人博客（PersonalBlog）系统，旨在建立一个小型的个人博客，可以分享自己的个人动态，也可浏览其他人的动态。

##### 3、软件定义

##### 4、主要仪器设备及耗材：

myeclipse10,mysql数据库，win10系统，

### 二、需求分析（问题陈述、需完成的功能。）

##### 1、问题陈述

个人博客最主要的目的是分享动态信息，而其又是一个单独的新的账户体系，不与当前最为流行的几大社交软件账户通用，因此，用户需得浏览其他用户的信息，以此达到交流的目的，若遇到感兴趣的文章或用户，便可以评论和点赞，关注。同时，用户也可以查看其粉丝的动态。同时，为避免不良信息的传播，管理员应当拥有封禁功能。同时，用户的注册，登录也不可少。

##### 2、功能需求

2.1 访客：

* 注册新用户
* 浏览博客主页
* 点赞

2.2一般用户：

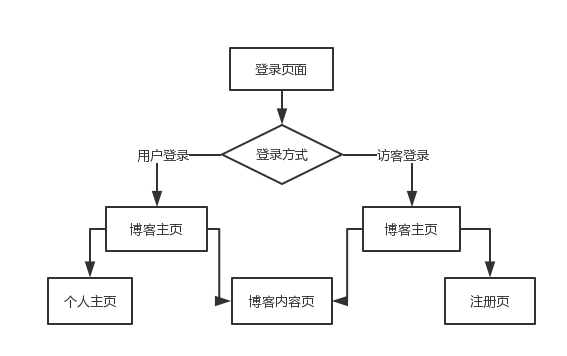
* 登录
* 浏览博客主页
* 点赞，关注，评论
* 浏览个人主页（查看我的好友 ，我的粉丝，我的评论）
* 创建新博客

2.3 管理员用户

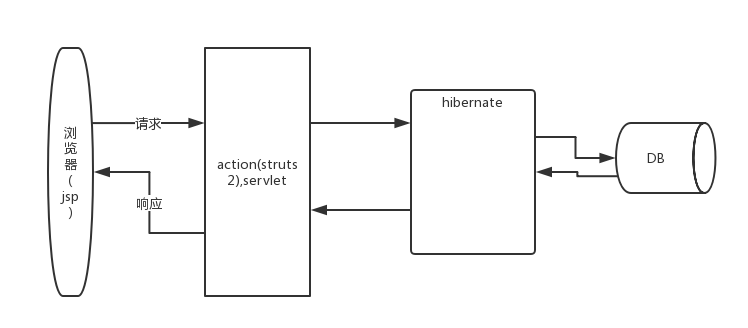
* 封禁博客
* 一般用户功能

### 三、系统设计（可用UML建模）

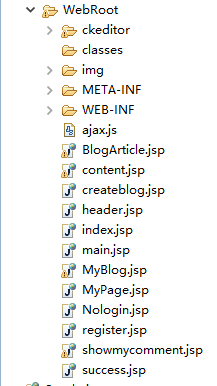
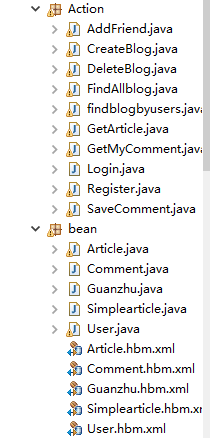
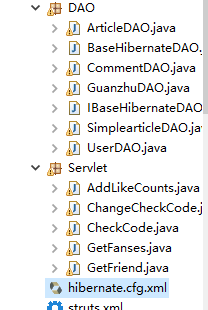
个人博客作为一个信息分享平台，主要是为用户提供一个渠道去分享用户的动态，以及浏览其他用户的动态。在用户进入网站首页时，可以有两个选择，一是选择用户登录，二是访客登录。用户登录后可进入网站主页，显示账户（启发用户所发布的），发表评论，关注用户，和为文章点赞。也可抵达个人主页，查看自己以前的博客及创建新博客。访客又可选择注册。但没有权限进入个人主页及编辑新博客。另，用户账户ID为10000以内的是自留的管理员账号，可封禁他人博客。



结构图：



ps：系统结构目录图：

数据库实现在文件MySQLcreate.txt中

### 四、程序原代码及其说明

组件图：

web.xml:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app version=*"2.5"*

xmlns=*"http://java.sun.com/xml/ns/javaee"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee*

*http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*>

<display-name></display-name>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<servlet>

<description>This is the description of my J2EE component</description>

<display-name>This is the display name of my J2EE component</display-name>

<servlet-name>GetFriend</servlet-name>

<servlet-class>Servlet.GetFriend</servlet-class>

</servlet>

<servlet>

<description>This is the description of my J2EE component</description>

<display-name>This is the display name of my J2EE component</display-name>

<servlet-name>AddLikeCounts</servlet-name>

<servlet-class>Servlet.AddLikeCounts</servlet-class>

</servlet>

<servlet>

<description>This is the description of my J2EE component</description>

<display-name>This is the display name of my J2EE component</display-name>

<servlet-name>GetFanses</servlet-name>

<servlet-class>Servlet.GetFanses</servlet-class>

</servlet>

<servlet>

<description>This is the description of my J2EE component</description>

<display-name>This is the display name of my J2EE component</display-name>

<servlet-name>ChangeCheckCode</servlet-name>

<servlet-class>Servlet.ChangeCheckCode</servlet-class>

</servlet>

<servlet>

<servlet-name>CheckCode</servlet-name>

<servlet-class>Servlet.CheckCode</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ChangeCheckCode</servlet-name>

<url-pattern>/changecheckcode</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>CheckCode</servlet-name>

<url-pattern>/checkcode</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>GetFriend</servlet-name>

<url-pattern>/getfriend</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AddLikeCounts</servlet-name>

<url-pattern>/addlikecounts</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>GetFanses</servlet-name>

<url-pattern>/getfanses</url-pattern>

</servlet-mapping>

<filter>

<filter-name>struts2</filter-name>

<filter-class>

org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter

</filter-class>

</filter>

<filter-mapping>

<filter-name>struts2</filter-name>

<url-pattern>\*.action</url-pattern>

</filter-mapping>

</web-app>

1.数据类设计：

* user.java

**package** bean;

**import** java.util.HashSet;

**import** java.util.Set;

/\*\*

\* User entity. **@author** MyEclipse Persistence Tools

\*/

**public** **class** User **implements** java.io.Serializable {

// Fields

**private** Integer id;

**private** String username;

**private** String password;

**private** Set simplearticles = **new** HashSet(0);

**private** Set guanzhus = **new** HashSet(0);

// Constructors

/\*\* default constructor \*/

**public** User() {

}

/\*\* full constructor \*/

**public** User(String username, String password, Set simplearticles,

Set guanzhus) {

**this**.username = username;

**this**.password = password;

**this**.simplearticles = simplearticles;

**this**.guanzhus = guanzhus;

}

// Property accessors

**public** Integer getId() {

**return** **this**.id;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

**public** String getUsername() {

**return** **this**.username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** **this**.password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** Set getSimplearticles() {

**return** **this**.simplearticles;

}

**public** **void** setSimplearticles(Set simplearticles) {

**this**.simplearticles = simplearticles;

}

**public** Set getGuanzhus() {

**return** **this**.guanzhus;

}

**public** **void** setGuanzhus(Set guanzhus) {

**this**.guanzhus = guanzhus;

}

}

* Simplearticle.java(博客简介)

**package** bean;

**import** java.util.Date;

**public** **class** Simplearticle {

**private** Integer articleId; //博客ID，主键

**private** String title; //标题

**private** String introdution; //简介

**private** Date updateTime; //上传时间

**private** User user; //发布者

**private** Article article; //博客主体

**private** Integer like\_count; //点赞数

**public** Article getArticle() {

**return** article;

}

**public** **void** setArticle(Article article) {

**this**.article = article;

}

**public** User getUser() {

**return** user;

}

**public** **void** setUser(User user) {

**this**.user = user;

}

**public** Integer getArticleId() {

**return** articleId;

}

**public** **void** setArticleId(Integer articleId) {

**this**.articleId = articleId;

}

**public** Date getUpdateTime() {

**return** updateTime;

}

**public** **void** setUpdateTime(Date updateTime) {

**this**.updateTime = updateTime;

}

**public** String getTitle() {

**return** title;

}

**public** **void** setTitle(String title) {

**this**.title = title;

}

**public** String getIntrodution() {

**return** introdution;

}

**public** **void** setIntrodution(String introdution) {

**this**.introdution = introdution;

}

**public** Simplearticle(){}

**public** Integer getLike\_count() {

**return** like\_count;

}

**public** **void** setLike\_count(Integer like\_count) {

**this**.like\_count = like\_count;

}

}

* article.java(博客内容)

package bean;

import java.awt.Image;

import java.util.List;

public class Article {

private Integer articleId;

private String content;

//private List<Image> images;

private Simplearticle saticle;

public Integer getArticleId() {

return articleId;

}

public void setArticleId(Integer articleId) {

this.articleId = articleId;

}

public Simplearticle getSaticle() {

return saticle;

}

public void setSaticle(Simplearticle saticle) {

this.saticle = saticle;

}

public String getContent() {

return content;

}

public void setContent(String content) {

this.content = content;

}

public Article(){}

}

* Guanzhu.java(关注的人)

**package** bean;

**public** **class** Guanzhu {

**private** Integer guanzhuId;

**private** Integer firendid;

**private** String firendName;

**private** User user;

**public** Integer getGuanzhuId() {

**return** guanzhuId;

}

**public** **void** setGuanzhuId(Integer guanzhuId) {

**this**.guanzhuId = guanzhuId;

}

**public** User getUser() {

**return** user;

}

**public** **void** setUser(User user) {

System.*out*.println("进入设置User");

**this**.user = user;

}

**public** Integer getFirendid() {

**return** firendid;

}

**public** **void** setFirendid(Integer firendid) {

**this**.firendid = firendid;

}

**public** String getFirendName() {

**return** firendName;

}

**public** **void** setFirendName(String firendName) {

**this**.firendName = firendName;

}

}

* Comment.java(评论)

**package** bean;

**public** **class** Comment {

**private** Integer commentId;

**private** Simplearticle simplearticle;

**private** User user;

**private** String comment;

**public** Integer getCommentId() {

**return** commentId;

}

**public** **void** setCommentId(Integer commentId) {

**this**.commentId = commentId;

}

**public** Simplearticle getSimplearticle() {

**return** simplearticle;

}

**public** **void** setSimplearticle(Simplearticle simplearticle) {

System.*out*.println("设置simplearticle");

**this**.simplearticle = simplearticle;

}

**public** User getUser() {

**return** user;

}

**public** **void** setUser(User user) {

**this**.user = user;

}

**public** String getComment() {

**return** comment;

}

**public** **void** setComment(String comment) {

**this**.comment = comment;

}

**public** Comment(){}

}

2.Hibernate设计：

* 配置文件hibernate.cfg.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"dialect"*>

org.hibernate.dialect.MySQLMyISAMDialect

</property>

<property name=*"connection.url"*>

jdbc:mysql://localhost:3306/Blog

</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*>123456</property>

<property name=*"connection.driver\_class"*>

com.mysql.jdbc.Driver

</property>

<mapping resource=*"bean/User.hbm.xml"* />

<mapping resource=*"bean/Guanzhu.hbm.xml"* />

<mapping resource=*"bean/Article.hbm.xml"* />

<mapping resource=*"bean/Simplearticle.hbm.xml"* />

<mapping resource=*"bean/Comment.hbm.xml"* />

</session-factory>

</hibernate-configuration>

* 映射文件User.hbm.xml

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<!--

Mapping file autogenerated by MyEclipse Persistence Tools

-->

<hibernate-mapping>

<class name=*"bean.User"* table=*"user"* catalog=*"blog"*>

<id name=*"id"* type=*"java.lang.Integer"*>

<column name=*"id"* />

<generator class=*"identity"* />

</id>

<property name=*"username"* type=*"java.lang.String"*>

<column name=*"username"* length=*"16"* />

</property>

<property name=*"password"* type=*"java.lang.String"*>

<column name=*"password"* length=*"20"* />

</property>

<set name=*"simplearticles"* inverse=*"true"*>

<key>

<column name=*"userId"* />

</key>

<one-to-many class=*"bean.Simplearticle"* />

</set>

<set name=*"guanzhus"* inverse=*"true"*>

<key>

<column name=*"id"* />

</key>

<one-to-many class=*"bean.Guanzhu"* />

</set>

</class>

</hibernate-mapping>

* 映射文件Simplearticle.hbm.xml

<hibernate-mapping package=*"bean"*>

<class name=*"Simplearticle"*>

<id name=*"articleId"* type=*"integer"*>

<generator class=*"identity"*></generator>

</id>

<property name=*"title"* type=*"string"*></property>

<property name=*"introdution"* type=*"string"*></property>

<property name=*"updateTime"* type=*"date"*></property>

<property name=*"like\_count"* type=*"integer"*></property>

<one-to-one name=*"article"*></one-to-one>

<many-to-one name=*"user"* column=*"userId"* lazy=*"false"*></many-to-one>

</class>

</hibernate-mapping>

* 映射文件Article.hbm.xml

<hibernate-mapping package=*"bean"*>

<class name=*"Article"*>

<id name=*"articleId"*>

<generator class=*"foreign"*>

<param name=*"property"*>saticle</param>

</generator>

</id>

<property name=*"content"*></property>

<one-to-one name=*"saticle"* constrained=*"true"*></one-to-one>

</class>

</hibernate-mapping>

* 映射文件Guanzhu.hbm.xml

<hibernate-mapping package=*"bean"*>

<class name=*"Guanzhu"*>

<id name=*"guanzhuId"*>

<generator class=*"identity"*></generator>

</id>

<property name=*"firendid"*></property>

<property name=*"firendName"*></property>

<many-to-one name=*"user"* column=*"id"* lazy=*"false"*></many-to-one>

</class>

</hibernate-mapping>

* 映射文件Comment.hbm.xml

<hibernate-mapping package=*"bean"*>

<class name=*"Comment"*>

<id name=*"commentId"*>

<generator class=*"identity"*>

</generator>

</id>

<property name=*"comment"*></property>

<many-to-one name=*"simplearticle"* column=*"articleId"* ></many-to-one>

<many-to-one name=*"user"* column=*"userId"* ></many-to-one>

</class>

</hibernate-mapping>

3.struts2,servlet和jsp：

* Struts配置文件struts.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE struts PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"

"http://struts.apache.org/dtds/struts-2.3.dtd">

<struts>

<package name=*"default"* namespace=*"/"* extends=*"struts-default"*>

<action name=*"login"* class=*"Action.Login"*>

<result name=*"success"*>/main.jsp</result>

<result name=*"loginerror"*>/index.jsp</result>

</action>

<action name=*"createblog"* class=*"Action.CreateBlog"*>

<result name=*"success"* type=*"chain"*>findmyblog</result>

</action>

<action name=*"findmyblog"* class=*"Action.findblogbyusers"*>

<result name=*"success"*>/MyBlog.jsp</result>

<result name=*"nologin"*>/Nologin.jsp</result>

</action>

<action name=*"findallblog"* class=*"Action.FindAllblog"*>

<result name=*"success"*>/content.jsp</result>

</action>

<action name=*"getarticle"* class=*"Action.GetArticle"*>

<result name=*"success"*>/BlogArticle.jsp</result>

</action>

<action name=*"savecomment"* class=*"Action.SaveComment"*>

<result name=*"success"* type=*"chain"*>getarticle</result>

</action>

<action name=*"deleteblog"* class=*"Action.DeleteBlog"*>

<result name=*"success"* type=*"chain"*>findallblog</result>

</action>

<action name=*"addfriend"* class=*"Action.AddFriend"*>

<result name=*"success"*>/BlogArticle.jsp</result>

<result name=*"error"*>/BlogArticle.jsp</result>

</action>

<action name=*"register"* class=*"Action.Register"*>

<result name=*"success"*>/success.jsp</result>

</action>

<action name=*"getmycomment"* class=*"Action.GetMyComment"*>

<result name=*"success"*>/showmycomment.jsp</result>

</action>

</package>

</struts>

* 登陆功能
  + - * index.jsp(登录页面)

<head>

<base href=*"*<%=basePath%>*"*>

<style type=*"text/css"*>

*.main*{

text-align: *center*; /\*让div内部文字居中\*/

background-color: *#ededed*;

border-radius: *20px*;

width: *300px*;

height: *350px*;

margin: *auto*;

position: *absolute*;

top: *0*;

left: *0*;

right: *0*;

bottom: *0*;

}

**p**{

font-size:*30px*;

color:*#337abf*;

margin:*0*;

}

*.login*{

border-radius: *4px*;

border-width:*1px*;

border-color:*#cbcbcb*;

margin-top:*30px*;

margin-left:*20px*;

height:*30px*;

}

*#submit*{

margin-top:*30px*;

height:*50px*;

width:*230px*;

border-width:*2px*;

border-color:*#cbcbcb*;

border-radius: *8px*;

background-color: *#337abf*;

color:*#fff*;

font-soze:*20px*;

}

**b**{

font-soze:*20px*;

color:*#337abf*;

}

**a**{

font-soze:*20px*;

color:*#337abf*;

text-decoration:*none*;

}

</style>

<title>My JSP 'index.jsp' starting page</title>

<meta http-equiv=*"pragma"* content=*"no-cache"*>

<meta http-equiv=*"cache-control"* content=*"no-cache"*>

<meta http-equiv=*"expires"* content=*"0"*>

<meta http-equiv=*"keywords"* content=*"keyword1,keyword2,keyword3"*>

<meta http-equiv=*"description"* content=*"This is my page"*>

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

</head>

<body bgcolor=*"#ededed"*>

<%session.setAttribute("iflogin", **false**);

session.setAttribute("User",**null**); %>

<div class=*"main"*>

<p align=*"center"* >MyBlog</p>

<form action=*"login.action"* method=*"post"*>

<b>账户</b><input type=*"text"* name=*"user.id"* class=*"login"*><br>

<b>密码</b><input type=*"password"* name=*"user.password"* class=*"login"*><br>

<b>验证码：</b><input type=*"text"* name=*"checkcode"* class=*"login"*/>

<img border=*"0"* src=*"checkcode"*/>

<input type=*"submit"* value=*"换一张"* onclick="form1.action='changecheckcode'"><br>

<input type=*"submit"* value=*"登录"* id=*"submit"*><br>

<a href=*"main.jsp"*>访客登录</a><br>

</form>

</div>

<%String message=(String)request.getAttribute("errormessage");

**if**(message==**null**) {}

**else**{

%>

<p><%= message%></p>

<% }%>

</body>

* + - * Login.java（验证登录的action）

**public** **class** Login **extends** ActionSupport {

**private** User user;

**private** HttpServletRequest request;

**private** HttpSession session;

**public** User getUser() {

**return** user;

}

**public** **void** setUser(User user) {

**this**.user = user;

}

**public** Login(){

request=ServletActionContext.*getRequest*();

session=request.getSession();

}

**public** String execute() **throws** Exception{

UserDAO userdao=**new** UserDAO();

System.*out*.println(user.getId());

String checkcode=request.getParameter("checkcode");

String truecheckcode=(String)session.getAttribute("checkCode");

**if**(!truecheckcode.equals(checkcode))

{

System.*out*.println(truecheckcode+checkcode );

request.setAttribute("errormessage", "验证码错误");

**return** "loginerror";

}

**if**(user.getId()==**null**)

{

request.setAttribute("errormessage", "账户不能为空");

**return** "loginerror";

}

User response=userdao.findById(user.getId());

**if**(response==**null**)

{

request.setAttribute("errormessage", "没有该账户");

**return** "loginerror";

}

**if**(response.getPassword().equals(user.getPassword()))

{

session.setAttribute("iflogin", **true**);

session.setAttribute("User", response);

**return** "success";

}

request.setAttribute("errormessage", "密码错误");

**return** "loginerror";

}

}

* 浏览主页
* ajax.js(AJax技术js)

**var** xmlHttpRequest=**null**;

**function** createXHR(){

**try**{

xmlHttpRequest =**new** XMLHttpRequest();

}**catch**(e1)

{}

**var** \_msxmlhttp =**new** Array("Msxml2.XMLHTTP6.0","Msxml2.XMLHTTP5.0","Msxml2.XMLHTTP4.0","Msxml2.XMLHTTP3.0","Msxml2.XMLHTTP","Microsoft.XMLHTTP");

**for**(**var** i=0;i<\_msxmlhttp.length;i++){

**try**

{

xmlHttpRequest=**new** ActiveXObject(\_msxmlhttp[i]);

**if**(xmlHttpRequest!=**null**)

{**break**;}

}**catch**(e2){}

}

**if**(xmlHttpRequest==**null**)

{alert("不能创建");

}

}

**function** sendRequest(url,params,method,handler){

createXHR();

**if**(!xmlHttpRequest) **return** **false**;

**if**(method=="GET")

{

xmlHttpRequest.open(method, url+'?'+params, **true**);

xmlHttpRequest.send(**null**);

}

**if**(method=="POST")

{

xmlHttpRequest.open(method, url, **true**);

xmlHttpRequest.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

xmlHttpRequest.send(params);

}

xmlHttpRequest.onreadystatechange =handler;

}

* main.jsp(框架设计页面)

<frameset rows=*"80,\*"* id=*"parent"* frameborder=*"0"* >

<frame src=*"header.jsp"* scrolling=*"no"* bordercolor=*""*>

<frameset cols=*"300,\*"*>

<%**if**((Boolean)session.getAttribute("iflogin")) {%>

<frame src=*"MyPage.jsp"* name=*"right"* scrolling=*"no"*>

<%}

**else**{ %>

<frame src=*"register.jsp"* name=*"right"* scrolling=*"no"*>

<%} %>

<frame src=*"findallblog.action"* scrolling=*"auto"* name=*"main"*>

</frameset>

</frameset>

* header.jsp(主页面头部设计)

<style type=*"text/css"*>

*#nav* { width:*100%*; height: *76px*; border: *1px solid #D4CD49*; position:*fixed*;left:*0*;top:*0* }

**td**{

height:*80px*;

font-size:*24px*;

padding:*40px*;

}

**a**{

text-decoration:*none*;

}

</style>

</head>

<body bgcolor=*" #ededed"*>

<div id=*"nav"* align=*"center"*>

<table>

<tr height=*"80px"*>

<td ><a href=*"findallblog.action"* target=*"main"*>推荐博客</a></td>

<td><a href=*"findmyblog.action"* target=*"main"*>我的博客</a></td>

</tr>

</table>

</div>

</body>

* myPage.jsp（个人信息页面,登陆后出现在主页面左侧）

<style type=*"text/css"*>

*#nav* { width:*270px*; height: *400px*; border: *0px solid #000*; position:*fixed*;left:*30px*;top:*30px*;scrolling:*yes*;backgroundcolor=" #fff" }

*#comment*{

text-decoration:*none*;

font-color:*#000*;

}

</style>

<script type=*"text/javascript"* src=*"ajax.js"*></script>

<script type=*"text/javascript"*>

**function** GetFriends(){

**var** url="getfriend";

**var** params="";

clear();

sendRequest(url,params,'POST',ShowFriends);

}

**function** GetFanses(){

**var** url="getfanses";

**var** params="";

clearfanses();

sendRequest(url,params,'POST',ShowFanses);

}

**function** ShowFriends(){

**if**(xmlHttpRequest.readyState==4){

**if**(xmlHttpRequest.status==200){

document.getElementById("friends").style.visibility="visible";

**var** tbody=document.getElementById("friendtbody");

**var** friendslist=**null**;

friendslist=xmlHttpRequest.responseText.split(",");

**var** friendsnum=friendslist.length;

console.log(friendsnum);

**for**(i=0;i<friendsnum;i++)

{

**var** newTr=document.createElement('tr');

**var** newCell=document.createElement('td');

**var** wordText=friendslist[i];

console.log(friendslist[i]);

**var** textNode=document.createTextNode(wordText);

newCell.onmouseover=setBgColor;

newCell.onclick=getBlogs;

newCell.appendChild(textNode);

newTr.appendChild(newCell);

tbody.appendChild(newTr);

}

}

}

}

**function** ShowFanses(){

**if**(xmlHttpRequest.readyState==4){

**if**(xmlHttpRequest.status==200){

document.getElementById("fanses").style.visibility="visible";

**var** tbody=document.getElementById("fansetbody");

**var** fanseslist=xmlHttpRequest.responseText.split(",");

**var** fansesnum=fanseslist.length;

**for**(i=0;i<fansesnum;i++)

{

**var** newTr=document.createElement('tr');

**var** newCell=document.createElement('td');

**var** textNode=document.createTextNode(fanseslist[i]);

newCell.onmouseover=setBgColor;

newCell.onclick=getBlogs;

newCell.append(textNode);

newTr.append(newCell);

tbody.append(newTr);

}

}

}

}

**function** clear(){

**var** tbody=document.getElementById("friendtbody");

**var** trs=tbody.getElementsByTagName('tr');

**for**(**var** i=trs.length-1;i>=0;i--)

{

tbody.removeChild(trs[i]);

}

}

**function** clearfanses(){

**var** tbody=document.getElementById("fansetbody");

**var** trs=tbody.getElementsByTagName('tr');

**for**(**var** i=trs.length-1;i>=0;i--)

{

tbody.removeChild(trs[i]);

}

}

**function** setBgColor(){

trSrc=event.srcElement;

**if**(trSrc)

trSrc.style.backgroundColor="white";

trSrc.style.backgroundColor="gray";

}

**function** getBlogs(){

**var** info=trSrc.firstChild.data;

**var** infolist=info.split("(");

**var** username=infolist[0];

**var** idre=infolist[1];

**var** id=idre.substr(0,idre.length-1);

alert(id);

window.parent.main.location.href="findmyblog.action?id="+id+"&name="+username+"&method=friends";

}

</script>

</head>

<body bgcolor=*" #ededed"* >

<% User user =(User)session.getAttribute("User");

String userinfo=user.getUsername()+"("+user.getId()+")"; %>

<div id=*"nav"* >

<p ><%=user.getUsername()%>(<%=user.getId()%>)<p>

<p id=*"Guanzhu"*><a onclick="GetFriends()">我的关注</a></p>

<div id=*"friends"* style="visibility:*hidden*" >

<table id=*"friendtable"*>

<tbody id=*"friendtbody"*><tr><td></td></tr></tbody>

</table>

</div>

<p><a onclick="GetFanses()">我的粉丝</a></p>

<div id=*"fanses"* style="visibility:*hidden*" >

<table id=*"fansetable"*>

<tbody id=*"fansetbody"*><tr><td></td></tr></tbody>

</table>

</div>

<p><a href=*"getmycomment.action"* target=*"main"* id=*"comment"*>我的评论</a></p>

</div>

</body>

* GetFriend.java(获取所有关注用户 servlet)

**public** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

HttpSession session=request.getSession();

User user =(User)session.getAttribute("User");

GuanzhuDAO gzdao=**new** GuanzhuDAO();

List friends=gzdao.findByProperty("user", user);

System.*out*.println("获取中"+friends.size()+user.getId());

Iterator it=friends.iterator();

String fname="";

**while**( it.hasNext() )

{

Guanzhu gz=(Guanzhu)it.next();

fname+=gz.getFirendName()+"("+gz.getFirendid()+")"+",";

System.*out*.println(fname);

}

response.setContentType("text;charset=UTF-8");

request.setCharacterEncoding("UTF-8");

PrintWriter out=response.getWriter();

out.print(fname);

}

* GetMyComment.java(获取你的所有评论action)

**public** **class** GetMyComment **extends** ActionSupport {

**private** HttpServletRequest request;

**private** HttpSession session;

**public** GetMyComment(){

request=ServletActionContext.*getRequest*();

session=request.getSession();

}

**public** String execute() **throws** Exception{

CommentDAO commentdao=**new** CommentDAO();

User user=**new** User();

user=(User)session.getAttribute("User");

List commentlist=commentdao.findByProperty("user", user);

request.setAttribute("comments", commentlist);

**return** "success";

}

}

* GetFanse.java(获取所有关注了你的用户 action)

**public** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

HttpSession session=request.getSession();

User user =(User)session.getAttribute("User");

GuanzhuDAO gzdao=**new** GuanzhuDAO();

List friends=gzdao.findByProperty("firendid", user.getId());

System.*out*.println("获取中"+friends.size());

Iterator it=friends.iterator();

String fname="";

**while**( it.hasNext() )

{

Guanzhu gz=(Guanzhu)it.next();

User ur=gz.getUser();

fname+=ur.getUsername()+"("+ur.getId()+")"+",";

System.*out*.println(fname);

}

response.setContentType("text;charset=UTF-8");

request.setCharacterEncoding("UTF-8");

PrintWriter out=response.getWriter();

out.print(fname);

}

* content.jsp(所有博客页面)

<style type=*"text/css"*>

*.sblog*{

}

</style>

</head>

<body>

<%

List<Simplearticle> salist=(ArrayList<Simplearticle>)session.getAttribute("salistall");

System.out.println("进入输出页面");

%>

<% Iterator it=salist.iterator();

**while**(it.hasNext()){

Simplearticle sa=(Simplearticle)it.next();

System.out.println(sa.getTitle());

**int** articleId=sa.getArticleId();

%>

<div class=*"sblog"* onclick=*"location.href='getarticle.action?articleId=*<%= articleId%>*';"*>

<h3><%= sa.getTitle()%></h3>

<p><%= sa.getUser().getUsername()%></p>

<p><%= sa.getIntrodution()%></p>

<p><%= sa.getUpdateTime()%></p>

<hr></hr>

</div>

<% }%>

</body>

</body>

</html>

* findallblog.java(查找所有博客的action)

**public** **class** FindAllblog **extends** ActionSupport {

**private** HttpServletRequest request;

**private** HttpSession session;

**public** FindAllblog(){

request=ServletActionContext.*getRequest*();

session=request.getSession();

}

**public** String execute() **throws** Exception{

System.*out*.println("进入fidallblogaction");

SimplearticleDAO sadao=**new** SimplearticleDAO();

List salist=sadao.findAll();

System.*out*.println(salist.size());

session.setAttribute("salistall", salist);

**return** "success";

}

}

* 查看博客
* BlogArticle.jsp

<script type=*"text/javascript"* src=*"ajax.js"*></script>

<script type=*"text/javascript"*>

**function** checkJurisdiction(){

console.log("asafasfas");

**var** isadmin=document.getElementById("button1");

console.log(isadmin.value);

**if**(isadmin.value>=10000){

window.alert("您没有权限进行此次操作！");

}

**else**{

window.location.href="deleteblog.action";

}

}

**function** Like(){

**var** img=document.getElementById("like");

**if**(img.getAttribute("src",2)=="img/like2.jpg")

{

**return**;

}

img.src="img/like2.jpg";

**var** url="addlikecounts";

**var** params="";

sendRequest(url,params,'GET',Like\_Count);

}

**function** Like\_Count(){

**if**(xmlHttpRequest.readyState==4)

{

**if**(xmlHttpRequest.status==200)

{

**var** likecounts=document.getElementById("likecounts");

**var** counts=xmlHttpRequest.responseText;

likecounts.innerHTML=counts;

}

}

}

</script>

<style type=*"text/css"*>

**h3**{

font:*italic bold 20px arial,sans-serif*;

}

</style>

</head>

<body>

<%Simplearticle sa=(Simplearticle)session.getAttribute("sa");

Article article=(Article)session.getAttribute("article");

User user=(User)session.getAttribute("User");

%>

<div id=*"article"*>

<h3><%= sa.getTitle()%></h3>

<p><%= sa.getIntrodution()%></p>

<p><%= sa.getUser().getUsername()%></p>

<p><%= sa.getUpdateTime()%></p>

<hr></hr>

<p><%= article.getContent()%></p>

<p align=*"right"*> <a onclick="Like()"><img id=*"like"* src=*"img/like1.jpg"* width=*"16"* height=*"16"*></img></a>

<b id=*"likecounts"*>${ sa.getLike\_count()}</b>

&nbsp;&nbsp;

<a href=*"addfriend.action"* >关注</a>

<button id=*"button1"* value=<%=user.getId() %> onclick="checkJurisdiction()">封禁</button>

</p>

</div>

<%

List<Comment> commentlist=(ArrayList<Comment>)session.getAttribute("commentlist");

System.out.println("进入评论输出页面");

%>

<% Iterator it=commentlist.iterator();

**while**(it.hasNext()){

Comment comment=(Comment)it.next();

%>

<div class=*"comment"*>

<hr></hr>

<h3><%= comment.getUser().getUsername()%></h3>

<p><%= comment.getComment()%></p>

</div>

<%} %>

<form action=*"savecomment.action"* method=*"post"*>

<textarea rows=*"4"* cols=*"80"* name=*"comment.comment"*></textarea>

<input type=*"hidden"* name=*"articleId"* value=<%= sa.getArticleId() %>>

<input type=*"submit"* value=*"评论"* />

</form>

</body>

* GetArticle.java(查看博客内容的action)

**public** **class** GetArticle **extends** ActionSupport {

**private** HttpServletRequest request;

**private** HttpSession session;

**public** GetArticle(){

request=ServletActionContext.*getRequest*();

session=request.getSession();

}

**public** String execute() **throws** Exception{

System.*out*.println("进入getablogaction");

String articleId=request.getParameter("articleId");

Integer articleid=Integer.*parseInt*(articleId);

System.*out*.println(articleId);

SimplearticleDAO sadao=**new** SimplearticleDAO();

ArticleDAO articledao=**new** ArticleDAO();

CommentDAO commentdao=**new** CommentDAO();

Simplearticle sa=sadao.findById(articleid);

Article article=articledao.findById(articleid);

List commentlist=commentdao.findByProperty("simplearticle", sa);

session.setAttribute("sa", sa);

session.setAttribute("article", article);

session.setAttribute("commentlist", commentlist);

**return** "success";

}

}

* AddFriend.java(添加关注的action)

**public** **class** AddFriend **extends** ActionSupport {

**private** HttpServletRequest request;

**private** HttpSession session;

**public** AddFriend(){

request=ServletActionContext.*getRequest*();

session=request.getSession();

}

**public** String execute() **throws** Exception{

Guanzhu guanzhu=**new** Guanzhu();

System.*out*.println("进入添加关注");

User user=(User)session.getAttribute("User");

guanzhu.setUser(user);

Simplearticle sa=(Simplearticle)session.getAttribute("sa");

guanzhu.setFirendid(sa.getUser().getId());

guanzhu.setFirendName(sa.getUser().getUsername());

GuanzhuDAO gzdao=**new** GuanzhuDAO();

**if**(gzdao.findexist(guanzhu).size()>0){

System.*out*.println(gzdao.findByExample(guanzhu).size());

**return** "error";

}

gzdao.save(guanzhu);

**return** "success";

}

}

* AddLikeCounts(添加点赞的servlet)

**public** **class** GetArticle **extends** ActionSupport {

**private** HttpServletRequest request;

**private** HttpSession session;

**public** GetArticle(){

request=ServletActionContext.*getRequest*();

session=request.getSession();

}

**public** String execute() **throws** Exception{

System.*out*.println("进入getablogaction");

String articleId=request.getParameter("articleId");

Integer articleid=Integer.*parseInt*(articleId);

System.*out*.println(articleId);

SimplearticleDAO sadao=**new** SimplearticleDAO();

ArticleDAO articledao=**new** ArticleDAO();

CommentDAO commentdao=**new** CommentDAO();

Simplearticle sa=sadao.findById(articleid);

Article article=articledao.findById(articleid);

List commentlist=commentdao.findByProperty("simplearticle", sa);

session.setAttribute("sa", sa);

session.setAttribute("article", article);

session.setAttribute("commentlist", commentlist);

**return** "success";

}

}

* SaveComment.java（添加评论的action）

**ublic** **class** SaveComment **extends** ActionSupport {

**private** Comment comment;

**private** HttpServletRequest request;

**private** HttpSession session;

**public** Comment getComment() {

**return** comment;

}

**public** **void** setComment(Comment comment) {

**this**.comment = comment;

}

**public** SaveComment(){

request=ServletActionContext.*getRequest*();

session=request.getSession();

}

**public** String execute() **throws** Exception{

System.*out*.println("添加评论");

Simplearticle sa=(Simplearticle)session.getAttribute("sa");

User user=(User)session.getAttribute("User");

System.*out*.println(sa.getTitle());

System.*out*.println(comment);

comment.setUser(user);

comment.setSimplearticle(sa);

CommentDAO commentdao=**new** CommentDAO();

commentdao.save(comment);

System.*out*.println("success");

**return** "success";

}

}

* 个人主页
* MyBlog.jsp(个人主页)

<style type=*"text/css"*>

**a**{

margin:*100px*;

text-decoration:*none*;

}

</style>

</head>

<body>

<h3><a href=*"findmyblog.action"* target=*"main"*>已发博客</a><a href=*"createblog.jsp"* target=*"main"*>新建博客</a></h3>

<%

List<Simplearticle> salist=(ArrayList<Simplearticle>)session.getAttribute("salist");

System.out.println("进入输出页面");

%>

<% Iterator it=salist.iterator();

**while**(it.hasNext()){

Simplearticle sa=(Simplearticle)it.next();

System.out.println(sa.getTitle());

**int** articleId=sa.getArticleId();

%>

<div onclick=*"location.href='getarticle.action?articleId=*<%= articleId%>*';"*>

<h3><%= sa.getTitle()%></h3>

<p><%= sa.getIntrodution()%></p>

<p><%= sa.getUpdateTime()%></p>

<hr></hr>

</div>

<% }%>

</body>

</html>

* createblog,jsp(创建博客页面)

<body>

<%**if**((Boolean)session.getAttribute("iflogin")){ %>

<form action=*"createblog.action"* method=*"post"*>

标题：<input type=*"text"* name=*"simplearticle.title"*><br>

简介：<input type=*"text"* name=*"simplearticle.introdution"*><br>

内容：<br>

<textarea class=*"ckeditor"* rows=*"10"* cols=*"80"* name=*"article.content"*></textarea><br>

<input type=*"submit"* value=*"提交"*>

</form>

<%}

**else** {%>

<p>您还没有登录，无法使用此功能，请先<a href=*"index.jsp"*>登录</a></p>

<%} %>

</body>

</html>

* createBlog.java(创建博客的action)

**public** **class** CreateBlog **extends** ActionSupport {

**private** Simplearticle simplearticle;

**private** Article article;

**private** HttpServletRequest request;

**private** HttpSession session;

**public** Simplearticle getSimplearticle() {

**return** simplearticle;

}

**public** **void** setSimplearticle(Simplearticle simplearticle) {

**this**.simplearticle = simplearticle;

}

**public** Article getArticle() {

**return** article;

}

**public** **void** setArticle(Article article) {

**this**.article = article;

}

**public** CreateBlog(){

request=ServletActionContext.*getRequest*();

session=request.getSession();

}

**public** String execute() **throws** Exception{

SimplearticleDAO sadao=**new** SimplearticleDAO();

ArticleDAO articledao=**new** ArticleDAO();

Date date=**new** Date();

System.*out*.println(date);

simplearticle.setUpdateTime(date);

User user=(User)session.getAttribute("User");

simplearticle.setUser(user);

List<Simplearticle> list=(List<Simplearticle>)sadao.findByProperty("user", user);

//simplearticle.setArticleId(list.size());

//System.out.println(simplearticle.getArticleId());

System.*out*.println(simplearticle.getTitle());

System.*out*.print(article.getContent());

sadao.save(simplearticle);

article.setSaticle(simplearticle);

articledao.save(article);

System.*out*.println("success");

**return** "success";

}

}

* findblogbyusers(获取已发博客)

**public** **class** findblogbyusers **extends** ActionSupport {

**private** HttpServletRequest request;

**private** HttpSession session;

**public** findblogbyusers(){

request=ServletActionContext.*getRequest*();

session=request.getSession();

}

**public** String execute() **throws** Exception{

System.*out*.println("进入action");

String method=request.getParameter("method");

User user=**new** User();

**if**(method==**null**)

{

user=(User)session.getAttribute("User");

System.*out*.println(user);

}

**else**

{

System.*out*.println("dianjileibiao");

String idstr=request.getParameter("id");

Integer id=Integer.*parseInt*(idstr);

String username=request.getParameter("name");

user.setId(id);

user.setUsername(username);

}

**if**(user==**null**)

{

**return** "nologin";

}

**else**

{

SimplearticleDAO sadao=**new** SimplearticleDAO();

List salist=sadao.findByProperty("user", user);

System.*out*.println(salist.size());

session.setAttribute("salist", salist);

**return** "success";

}

}

}

* 注册
* 注册页面register.jsp（访客登录时在主页面左侧显示）

<script type=*"text/javascript"*>

**function** validate(){

alert("确认注册？");

**var** name=document.form1.username.value;

**var** pwd=document.form1.pwd.value;

**var** pwd1=document.form1.password.value;

**if**(name.length<=0) alert("呢称不能为空！");

**else if**(pwd!=pwd1) alter("两次输入密码不一致！");

**else return true**;

}

</script>

<style type=*"text/css"*>

*#nav* { width:*268px*;left:*30px*; height: *400px*; border: *0px solid #D4CD49*; position:*fixed*;top:*0* }

*.ab*{margin:*20px 5px 10px 36px*;}

*#c*{margin:*20px 5px 10px 4px*;}

*#submit*{

margin:*10px 126px*;

}

</style>

</head>

<body bgcolor=*" #ededed"*>

<div id=*"nav"*>

<form action=*"register.action"* method=*"post"* name=*"form1"*>

<span>呢称：</span><input type=*"text"* class=*"ab"* name=*"username"*/> <br>

<span>密码：</span><input type=*"password"* class=*"ab"* name=*"pwd"*/> <br>

<span>确认密码：</span><input type=*"password"* name=*"password"* id=*"c"*/><br>

<input type=*"submit"* value=*"注册"* onclick=" return validate()" id=*"submit"*/ >

</form>

</div>

</body>

* CheckCode.java(获取验证码的servlet)

**public** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("image/jpeg");

HttpSession session=request.getSession();

**int** width=60;

**int** height=20;

//设置浏览器不要储存该图片

response.setHeader("Pragma", "No-cache");

response.setHeader("Cache-Control", "no-cache");

response.setDateHeader("Expires", 0);

//创建内存图像并获得图片上下文

BufferedImage image =**new** BufferedImage(width,height,BufferedImage.*TYPE\_INT\_RGB*);

Graphics g=image.getGraphics();

//产生随机的验证码

//定义验证码的字符表

String chars="0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";

**char**[] rands=**new** **char**[4];

**for**(**int** i=0;i<4;i++)

{

**int** rand=(**int**)(Math.*random*()\*36);

rands[i]=chars.charAt(rand);

}

//产生图像

//画背景

g.setColor(**new** Color(0xDCDCDC));

g.fillRect(0, 0, width, height);

//随机产生120个干扰点

**for**(**int** i=0;i<120;i++)

{

**int** x=(**int**)(Math.*random*()\*width);

**int** y=(**int**)(Math.*random*()\*height);

**int** red=(**int**)(Math.*random*()\*255);

**int** green=(**int**)(Math.*random*()\*255);

**int** blue=(**int**)(Math.*random*()\*255);

g.setColor(**new** Color(red,green,blue));

g.drawOval(x, y, 1, 0);

}

g.setColor(Color.*BLACK*);

g.setFont(**new** Font(**null**,Font.*ITALIC*|Font.*BOLD*,18));

//在不同的高度上输出不同的字符

g.drawString(""+rands[0], 1, 17);

g.drawString(""+rands[1], 16, 15);

g.drawString(""+rands[2], 31, 18);

g.drawString(""+rands[3], 46, 16);

g.dispose();

//将图像输出到客户端

ServletOutputStream sos=response.getOutputStream();

ByteArrayOutputStream baos=**new** ByteArrayOutputStream();

ImageIO.*write*(image, "JPEG", baos);

**byte**[] buffer=baos.toByteArray();

response.setContentLength(buffer.length);

sos.write(buffer);

baos.close();

sos.close();

//将验证码放入session中

session.setAttribute("checkCode", **new** String(rands));

}

* Register.java(注册的action)

**public** String execute() **throws** Exception{

User user=**new** User();

String username=request.getParameter("username");

user.setUsername(username);

String pwd=request.getParameter("pwd");

user.setPassword(pwd);

UserDAO userdao=**new** UserDAO();

userdao.save(user);

User response=userdao.findById(user.getId());

session.setAttribute("User", response);

**return** "success";

}

}

* 其他功能
* 管理员封禁功能DeleteBlog.java

**public** **class** DeleteBlog **extends** ActionSupport {

**private** HttpServletRequest request;

**private** HttpSession session;

**public** DeleteBlog(){

request=ServletActionContext.*getRequest*();

session=request.getSession();

}

**public** String execute() **throws** Exception{

Simplearticle sa=(Simplearticle)session.getAttribute("sa");

SimplearticleDAO sadao=**new** SimplearticleDAO();

Article a=sa.getArticle();

ArticleDAO adao=**new** ArticleDAO();

adao.delete(a);

sadao.delete(sa);

**return** "success";

}

}

### 五、总结

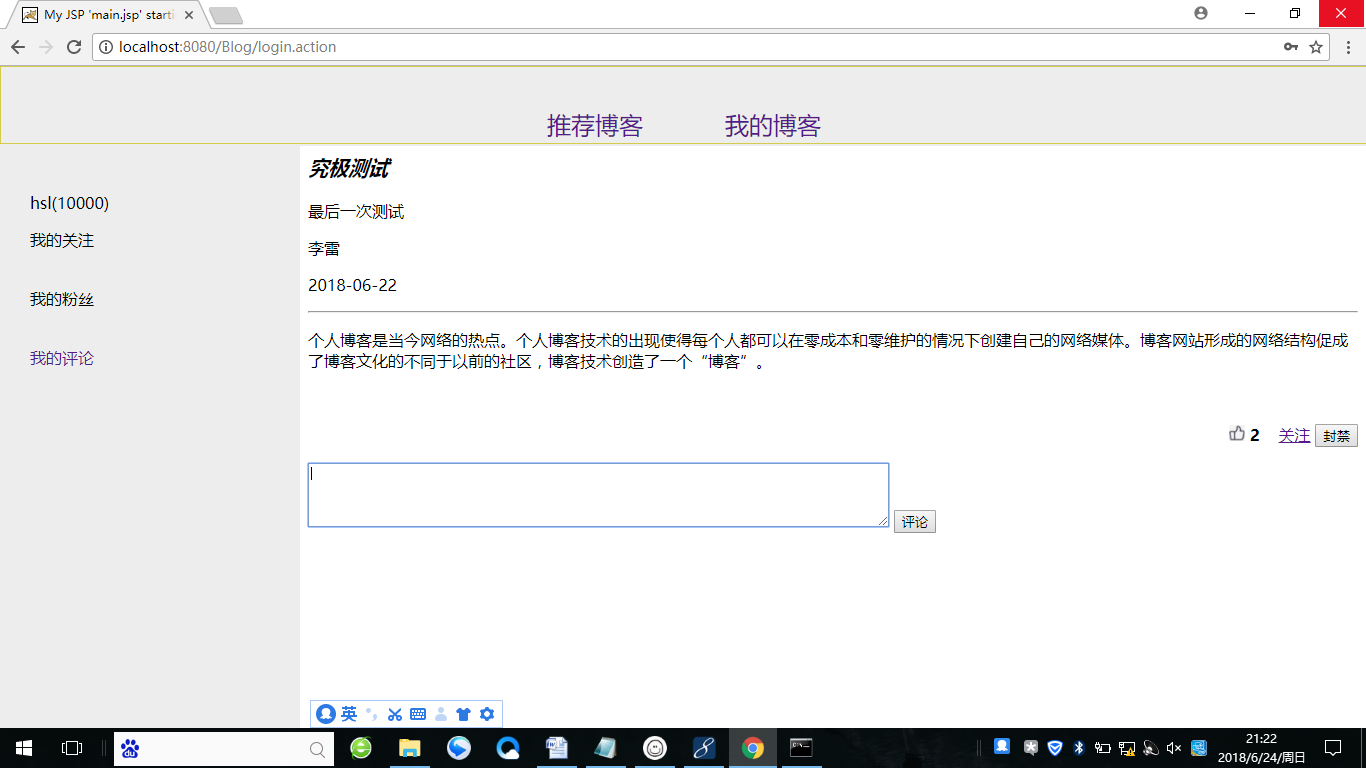
1.登录页面：



2.博客主页：



3.点击一个博客;



4.评论



5.关注并点赞



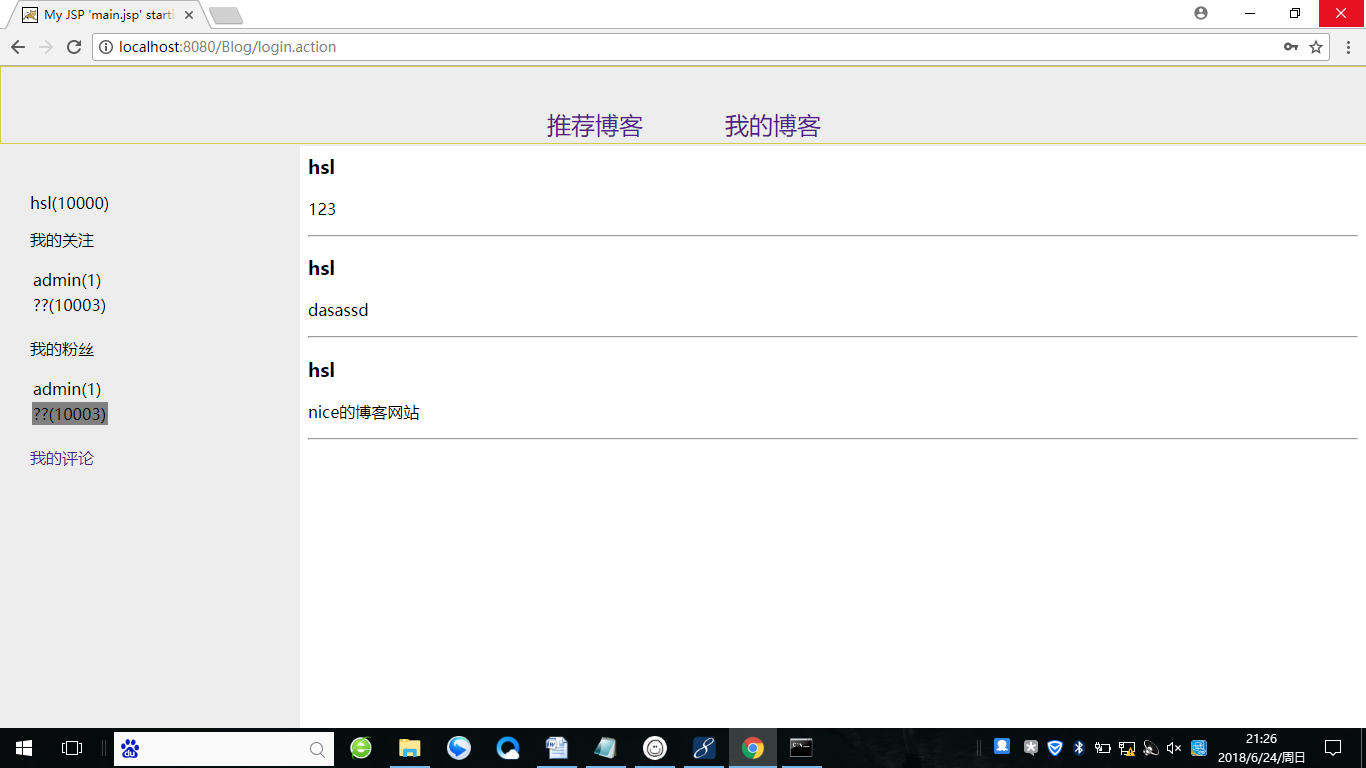
6.点击我的关注



7.点击我的粉丝



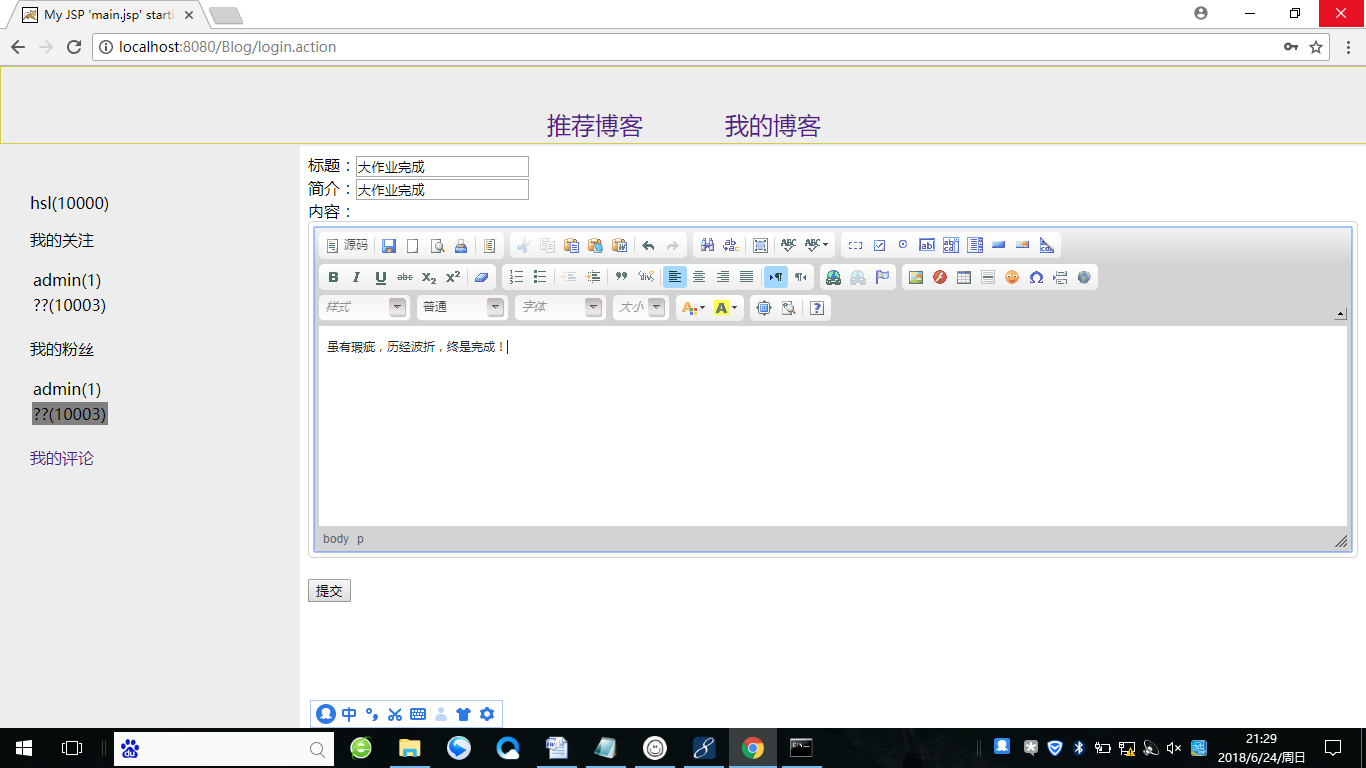
8.点击我的评论



9.点击我的博客



10.点击新建博客





12.点击一位关注或粉丝



13注册





13.管理员封禁(10000以内的用户登录)







### 六、参考文献

1.Java Web 应用开发技术与案例教程