

最终大作业说明

清单说明：

(WEB-INF:)	(js:)	(css:)
index.jsp(主页) ->index.html	ajax.js ->Ajax 应用技术	bootstrap.css
login_submit.jsp->登陆验证	jquery-3.2.1.min.js	bootstrap1.css
sign_up.html ->注册页	prototype-1.6.0.3.js 框架	bootstrapValidator.css
sign_up.jsp ->注册提交	login.js ->登陆	
main.html ->大厅页面	changetheme.js->切换主题	
login_db ->数据库文件	bootstrap 相关框架（包含表单验证）	

一、网站分区说明

网站共分为三个页面，分别是 index.html(主页)，main.html(主体部分)，sign_up.html(注册页面)，各个页面的结构框架如下图：
Index.html: 由 index.jsp 控制（导航栏只显示主页及注册和登陆及切换主题）



Main.html:



Sign_up.html:



二、设计说明

主页 (index.jsp)



大厅 (main.html)

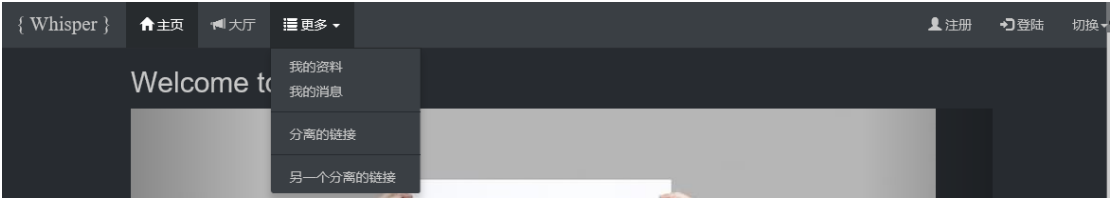


注册 (sign_up.html)

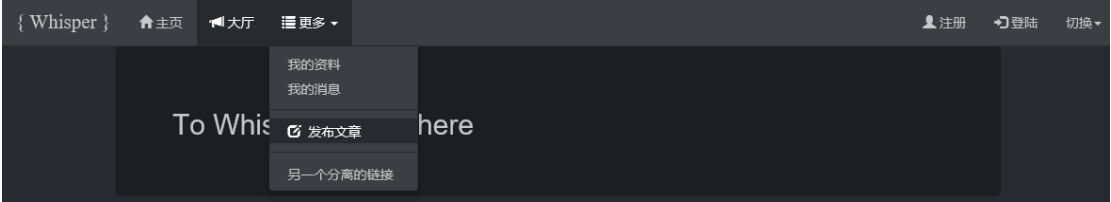


三个页面的导航栏基本一致,部分导航选项附加了字符图标,二级菜单为“更多”

和用于切换主题的“切换”选项，其中“更多”选项用于扩展其他功能



进入“大厅”选项下的“更多”选项中附有“发布文章”，可触发模态框：



每个导航栏的“登陆”选项用于弹出模态框进行用户登陆操作：



登陆示例：

登陆

用户名:

Takic

密码:

•••••

登陆成功, 三秒后自动跳转到: 我的大厅

取消

登陆

注册：实现对数据库表插入用户数据

表单验证：使用 bootstrap Validator 进行动态表单验证，由于时间关系，没有做到对重复用户名的验证，后期可根据 remote 远程应用 Action 在后台进行验证，表单对每一个表单元素设置了验证规则，以用户名验证为例如下图：

```
$(document).ready(function(){
    $('#signform').bootstrapValidator({
        message: 'This value is not valid',
        feedbackIcons: {
            valid: 'glyphicon glyphicon-ok',
            invalid: 'glyphicon glyphicon-remove',
            validating: 'glyphicon glyphicon-refresh'
        },
        fields: {
            username: {
                validators: {
                    notEmpty: {
                        message: '用户名不能为空'
                    },
                    stringLength: {
                        min: 3,
                        max: 22,
                        message: '用户名长度为3-22'
                    },
                    regexp: {
                        regexp: /^[0-9A-Za-z_]+$/,
                        message: '用户名由字母，数字及_组成'
                    },
                    different: {
                        field: 'password',
                        message: '用户名和密码不能相同'
                    }
                }
            },
            password: {
                validators: {
                    notEmpty: {
                        message: '密码不能为空'
                    },
                    stringLength: {
                        min: 6,
                        max: 18,
                        message: '密码长度为6-18'
                    },
                    regexp: {
                        regexp: /^[0-9A-Za-z_]{6,18}$/,
                        message: '密码由字母，数字及_组成'
                    },
                    different: {
                        field: 'username',
                        message: '用户名和密码不能相同'
                    }
                }
            }
        }
    });
});
```

表单在验证未通过时，不可点击“提交”进行 submit 的后续操作，验证无误后提交将跳转到 sign_up.jsp 进行数据库操作(如下图)，同时页面 2 秒后跳转回主页可进行登陆操作。

```
request.setCharacterEncoding("UTF-8");
String name=request.getParameter("username");
String password=request.getParameter("password");
String sex=request.getParameter("sex");
String grade=request.getParameter("grade");
String tel=request.getParameter("tel");
String comment=request.getParameter("comment");
String interest[]=request.getParameterValues("chk");
String ins="";
for(int i=0;i<interest.length;i++){
    ins=ins+interest[i]+" ";
}

DBConn db=new DBConn();
//验证用户是否已经被注册
/* if(db.checkUsername(name)==true){
    out.println("用户名已存在，请重新输入");
    return;
}else{ */
db.execSQL("insert into member(username,password,sex,grade,tel,interest,comment)values('"+name+"','"+password+"','"+sex+"','"+grade+"','"+
```

```
</>
<body onload="back()">
<br>
<p style="text-align:center">注册成功，返回: <a href=index.jsp>主页</a></p>
</body>
<script type="text/javascript">
function back(){
    setTimeout("window.location='index.jsp'",2000);
}
</script>
```

表单输入样例：

注册信息

用户名

45

✕

用户名长度为3-22

密码

●●●●●●

✓

确认密码

●●●●●●

✓

性别

男

▼

年级

大一

▼

专业兴趣

☐ 人工智能

☒ 数据挖掘

☒ 数学建模

☐ 网络编程

☐ 机器学习

☐ 多媒体技术

☐ 计算机图形学

☐ 移动平台开发

联系电话

asa

✕

该联系电话无效

联系电话长度为6-11

个人简介

输入您的个人介绍等附加信息

重置

提交

三、切换主题：

页面主题共两套 bootstrap.css(默认主题)以及 bootstrap1.css(白色简约)，基本组件是引用的是 bootstrap 框架的组件库（对颜色等效果稍作修改），利用触发“切换”的下拉选项进行切换，切换白色主题，字体切换为楷体，字号切换成 20px（原字体大小 14px），风格切换成白色系列。

1) 切换方法

```

11 <!--切换主题代码-->
12 <script type="text/javascript">
13     function change_default(){
14         var css=document.getElementById("css");
15         var _body=document.getElementById("_body");
16         css.href="css/bootstrap.css";
17         _body.style.fontFamily="";
18         _body.style.fontSize="";
19     }
20     function change_white(){
21         var css=document.getElementById("css");
22         var _body=document.getElementById("_body");
23         css.href="css/bootstrap1.css";
24         _body.style.fontFamily="楷体";
25         _body.style.fontSize="20px";
26     }
27     function loop(str){
28         var css=document.getElementById("css").href;
29         window.location.href=str+"?css="+css;//传递当前页面css样式表参数
30     }
31     function getParam(){
32         var css=document.getElementById("css");
33         var c1=window.location.href.split("?")[1];
34         var c2=c1.split("=")[1];
35         var _css=css.href.split('/');
36         if(css.href!=c2){
37             if(_css[_css.length-1]=="bootstrap.css")
38                 change_white();
39             else
40                 change_default();
41         }
42     };
43 </script>

```

```

<!--初始执行方法getParam()获取上一个页面的主题-->
<body id="_body" onLoad="getParam()">

```

页面一加载就执行 getParam() 方法获取 url，截取 loop(str) 方法传来的样式表参数，判断是否与前面的页面主题相同从而进行切换，每个导航栏上跳转页面的<a>标签都绑定一个带参 js 方法：

```

<li><a onClick="loop('index.html')" href="javascript:;"><span class="glyphicon glyphicon-home"></span> 主页</a></li>
<li><a onClick="loop('main.html')" href="javascript:;"><span class="glyphicon glyphicon-bullhorn"></span> 大厅</a></li>

```

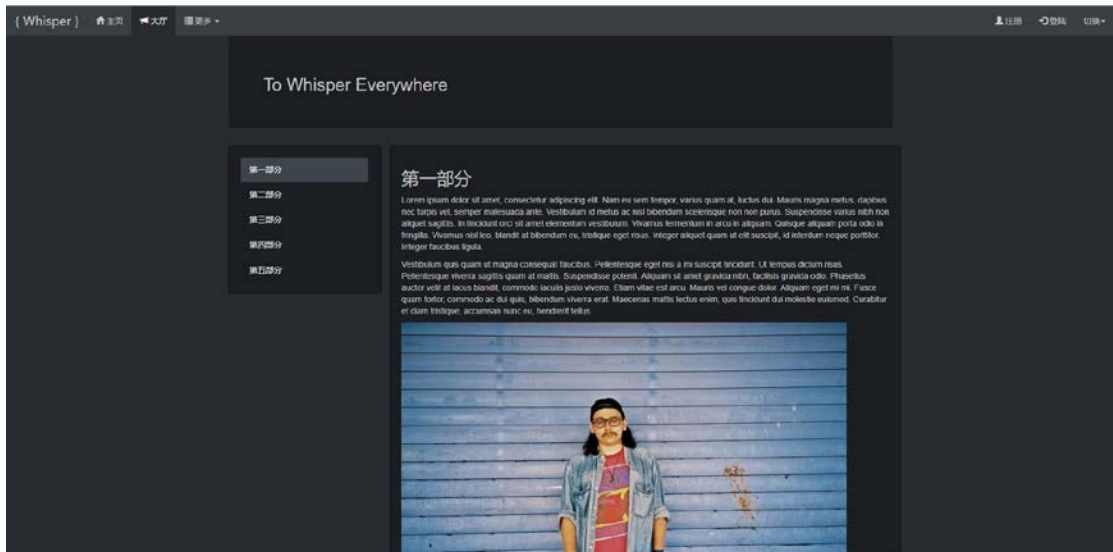
```

<li class="active"><a onClick="loop('sign_up.html')" href="javascript:;"><span class="glyphicon glyphicon-user"></span> 注册</a></li>

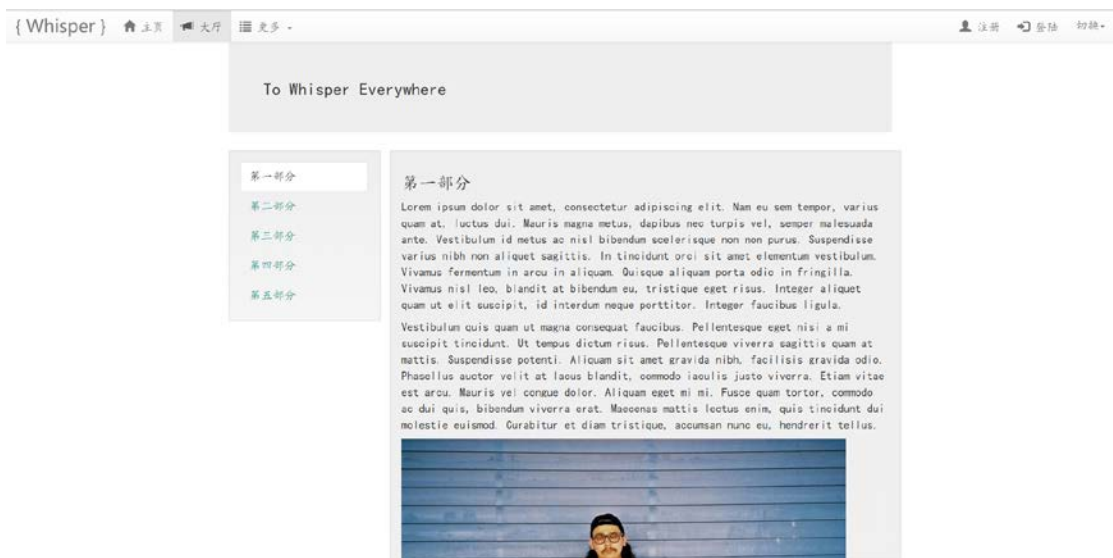
```

最终实现全部三个页面整体切换主题

切换前：



切换后：



2) 切换出现的问题

用 chrome, 360, qq 浏览器进行主题切换时, 跳转页面是会出现闪屏现象, 用 IE 以及 Microsoft edge 浏览器时不会出现闪屏, 后期可利用插入过渡效果减缓闪屏的影响。

四、Ajax 及 JSP 技术应用

主要是登陆验证时进行应用, Request 请求传参, 在 jsp 中对数据库进行操作, 从而验证是否登陆成功, 回调函数显示在登陆模态框中, 代码如下:

Js:

```
function login(){//登陆
    var pars="u="+$("#username").value+"&p="+$("#password").value;
    var myAjax=new Ajax.Request("login_submit.jsp",{method:"get",parameters:pars,onComplete:callLogin});
}

function callLogin(r){//回调函数
    var t=r.responseText.trim();
    var o=$("#acc");
    if(t=="ac"){
        o.innerHTML="登陆成功, 三秒后自动跳转到: <a href='main.html'>我的大厅</a>";
        setTimeout("window.location='main.html'",3000);
    }else{
        o.innerHTML=t;
    }
}
```


Jsp:

```
//获取各项参数值
String username=request.getParameter("u");//用户名
String password=request.getParameter("p");//用户密码
//验证用户名，三种情况：NULL;空;模式不匹配，需要重新填写
if(username==null||"".equals(username)||check("username",username)==false){
    out.println("用户名错误，请重新填写!");return;
}
//验证密码
if(password==null||"".equals(password)||check("password",password)==false){
    out.println("密码错误，请重新填写!");return;
}
DBConn db=new DBConn();
//验证用户名是否已经被注册
if(db.checkUsername(username)!=true){
    out.println("用户名不存在，请重试!");return;
}
boolean b=db.checkPassword(username, password);
if(b==true){
    session.setAttribute("uname",username);System.out.println("AAAAAAAAAAAAA");
    out.print("ac");
}else{
    out.print("登录失败，请重试!");
}
}
```

五、数据库应用

建立数据库 (login_db.txt):

```
DROP TABLE IF EXISTS `member`;
SET @saved_cs_client = @@character_set_client;
SET character_set_client = utf8;
CREATE TABLE `member` (
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `username` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  `sex` VARCHAR(45) DEFAULT NULL,
  `grade` VARCHAR(45) DEFAULT NULL,
  `tel` VARCHAR(45) DEFAULT NULL,
  `interest` VARCHAR(255) DEFAULT NULL,
  `comment` TEXT,
  PRIMARY KEY (`id`)
) ENGINE=INNODB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;
SET character_set_client = @saved_cs_client;

LOCK TABLES `member` WRITE;
/*!40000 ALTER TABLE `member` DISABLE KEYS */;
INSERT INTO `member` VALUES (1,'Takic','ttttt','男','大三','18750329229','Hello World!',NULL),(2,'John','jjjjj','男','大二','18275097079','Keep');
/*!40000 ALTER TABLE `member` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2018-01-01 20:29:40
```

建立数据库连接类:

```
public class DBConn {
    public Connection getConn(){
        Connection conn=null;
        try{
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            String url="jdbc:mysql://localhost:3307/whisper?user=root&password=962464&useUnicode=true&characterEncoding=UTF-8";
            conn=DriverManager.getConnection(url);
            return conn;
        }catch(Exception e){
            e.printStackTrace();
        }
        return conn;
    }

    public boolean checkUsername(String uname){//检查用户名是否存在
    }

    public boolean checkPassword(String uname,String pass){//检查密码是否正确
    }

    public int exeSQL(String sql){//执行更新语句
    }

    public Member getInfo(String uname){
    }
}
```

六、改进

此次大作业由于时间较紧，还有很多功能需要后期完善，对于表单的用户名查重验证、后期主功能区的实现、用户信息的显示等等较多问题，调试过程中出现 prototype 框架与表单的 Validator 验证相冲突，导致无法验证，切换主题时二级菜单不可下拉等等，前前后后自己琢磨了挺长时间才发现问题，在学习过程中领会了许多前后台交互的相关知识，受益颇深。