动态网站开发技术课程设计报告

**导言**

通过学习这门专业课程，我学会了怎样去搭建一个动态网站，在网页设计的基础上又迈进了一步。本次课程设计我的作品是一个简易的音乐网站，因为我平常喜欢听音乐，所以我想将自己喜欢的歌曲通过网站的形式展现出来，我将其命名为zyw音乐网。原打算做一个音乐播放平台，让用户能够通过账号登录，并可以将平台上播放的音乐喜欢的那部分收藏为自己的歌单，但由于时间及技术方面的因素，没有实现收藏歌单的功能。

**设计**

之前学过网页设计的课程，所以我对html、css、js可以比较熟练地使用，但本次实验是设计动态网站，这就涉及到了后端对数据库的操作，而我对c#操作数据库使用起来不熟练，所以本次实验主要是依赖王老师的博客，结合<http://nilnul.com/app_/nilnul._web_._KEN_/_http_/Lab.ascx>这篇文章我知道了在asp.net中前后端是如何交互的；结合<http://nilnul.com/_apps/wyt/LoadTag.aspx?url=/app_/nilnul.web.aspNet._Ken/app_/linq/Lab.ascx>这篇文章我学会了c#对数据库的基本操作。

**数据表**

设计音乐网站，首先是思考我做这个网站需要几张数据表，对于我需要的用户登录与歌单播放这两个功能，暂且只用得到两张表。

1. 使用VisualStudio创建Asp.Net WEB应用程序项目；
2. 在项目中添加SQL Server数据库并建立连接；
3. 新增表userInfo用户信息表、music用户歌曲表；

表 1 userInfo

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 允许NULL | 是否设为主键 |
| userId | int | FALSE | True |
| username | nvarchar(50) | FALSE | FALSE |
| password | nvarchar(50) | FALSE | FALSE |

表 2 music

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 允许NULL | 是否设为主键 |
| musicId | int | FALSE | True |
| userId | int | FALSE | FALSE |
| musicName | nvarchar(50) | FALSE | FALSE |
| musicAuthor | nvarchar(50) | FALSE | FALSE |
| musicImg | nvarchar(50) | FALSE | FALSE |
| musicUrl | nvarchar(50) | FALSE | FALSE |

1. 右键点击项目，新添一个Linq2SQL或Linq2Entity项目；
2. 把表格login转换为相应的类；
3. 设计页面，并代码实现；
4. Asp.Net处理页面请求。

**实验核心代码**

登录校验ajax请求：

$.ajax(

{

method: "get",

url: "/login.ashx",

data: {

"username": $("[name=username]").val(),

"password": $("[name=password]").val()

},

dataType: 'json',

contentType: 'application/x-www-form-urlencoded',

success: function (data) {

alert("登陆成功！");

localStorage.setItem("userInfo", JSON.stringify(data));

window.location.href = "/music";

},

error: function (xhr) {

alert("账号或密码错误！")

console.log(xhr.status);

}

}

)

Ajax请求歌曲信息：

var url = "/getMusic.ashx";

$.get(url, {

userId: JSON.parse(localStorage.getItem('userInfo')).userId

}, function (data) { //请求成功后的回调

//渲染播放控制台

var musicList = JSON.parse(data);

$('.r\_top>.title').text(musicList.musicName);

$('.r\_top>.autor').text("-" + musicList.musicAuthor);

$("#music").attr('src', musicList.musicUrl);

$('#music')[0].pause();

$('.left>.pic').attr('src', '/' + musicList.musicImg + '.jpg');

$("#username").text(JSON.parse(localStorage.getItem('userInfo')).username);

});

操作DOM H5的audio标签：

//点击播放按钮切换暂停按钮 播放暂停

$('.left>.iconfont').click(function () {

if ($('.left>.iconfont').hasClass('icon-bofang')) {

$('.left>.iconfont').removeClass('icon-bofang').addClass('icon-tubiaozhizuomoban');

$('#music')[0].play();

} else {

$('.left>.iconfont').addClass('icon-bofang').removeClass('icon-tubiaozhizuomoban');

$('#music')[0].pause();

}

});

//音乐播放时间变动时 进度条等渲染

$('#music').on('timeupdate', function () {

var bgSize = $('#music')[0].currentTime / $('#music')[0].duration \* 100 + '%';

$('.progress').css('background-size', bgSize);

$('.cur\_time').text(toTime($('#music')[0].currentTime));

});

//歌曲加载完之后 渲染歌曲的时长等

$('#music').on('canplay', function () {

durtime = $('#music')[0].duration;

$('.duration').text('/' + toTime(durtime));

});

//拖动进度条时

$('.progress').click(function (ev) {

var x = ev.clientX - $(this)[0].getBoundingClientRect().left;

$('#music')[0].currentTime = x / $(this).width() \* $('#music')[0].duration;

});

//时长秒转为 00:00格式

function toTime(num) {

var str = '';

var minute = parseInt(num / 60).toString().padStart(2, '0');

var seccond = parseInt(num % 60).toString().padStart(2, '0');

str = minute + ':' + seccond;

return str;

}

//00:00格式转为时长 秒

function toNum(str) {

var num = 0;

str.split(':').forEach(function (item, index, a) {

num = parseInt(a[0]) \* 60 + parseFloat(a[1]);

});

return num;

}

c#处理GET请求返回歌单信息：

class Data

{

public int musicId = 0;

public String musicAuthor = "";

public String musicName = "";

public String musicUrl = "";

public String musicImg = "";

}

public class getMusic : IHttpHandler

{

public void ProcessRequest(HttpContext context)

{

int userId = int.Parse(context.Request.QueryString["userId"]);

DataClasses1DataContext dc = new DataClasses1DataContext();

var data = new Data();

var q = from c in dc.music

where c.userId == userId

select c;

foreach (var item in q)

{

data.musicId = item.musicId;

data.musicAuthor = item.musicAuthor;

data.musicName = item.musicName;

data.musicUrl = item.musicUrl;

data.musicImg = item.musicImg;

}

JavaScriptSerializer jss = new JavaScriptSerializer();

string json = jss.Serialize(data);

context.Response.Write(json);

}

**效果图**

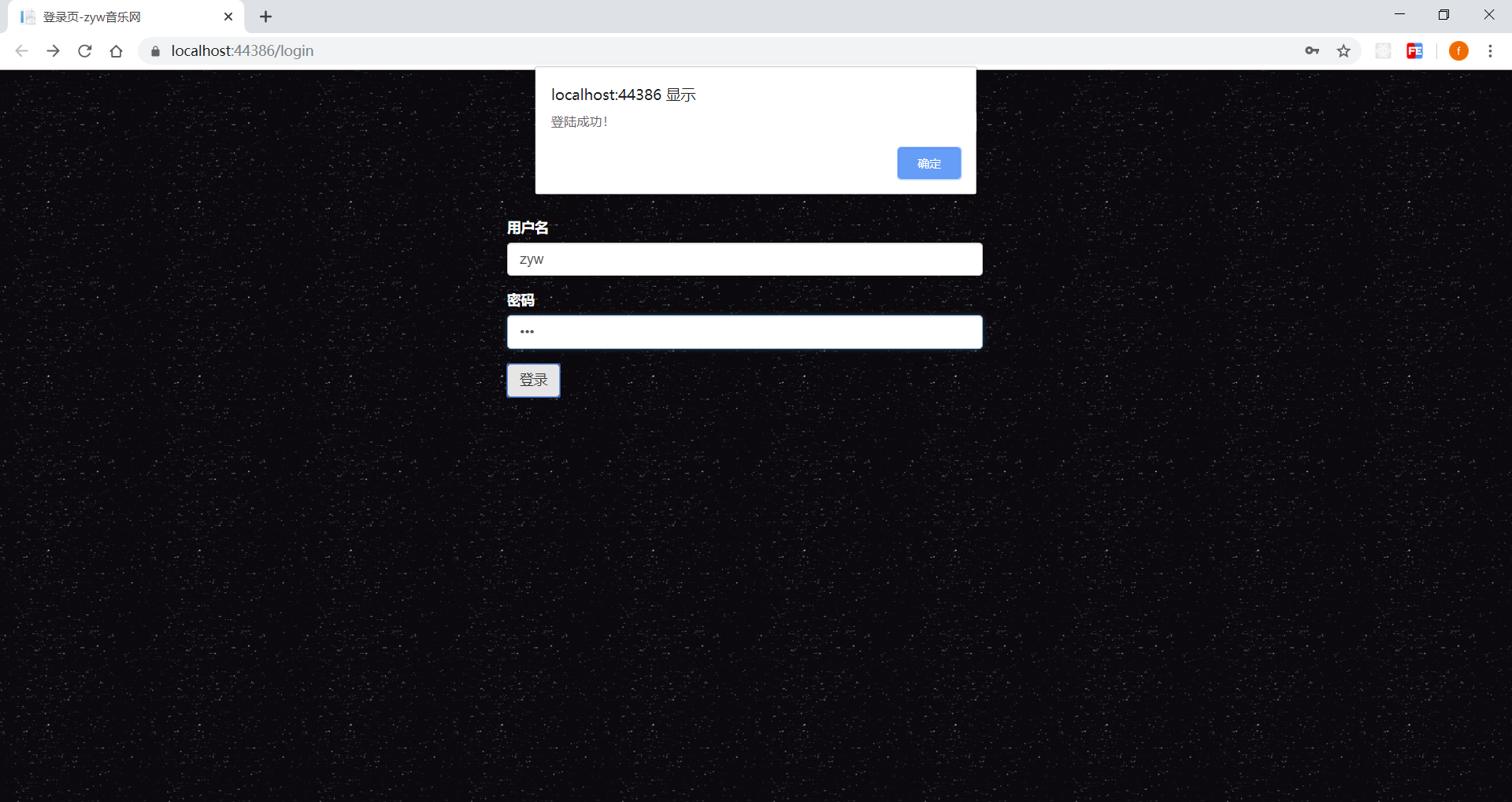


图 1 登录页

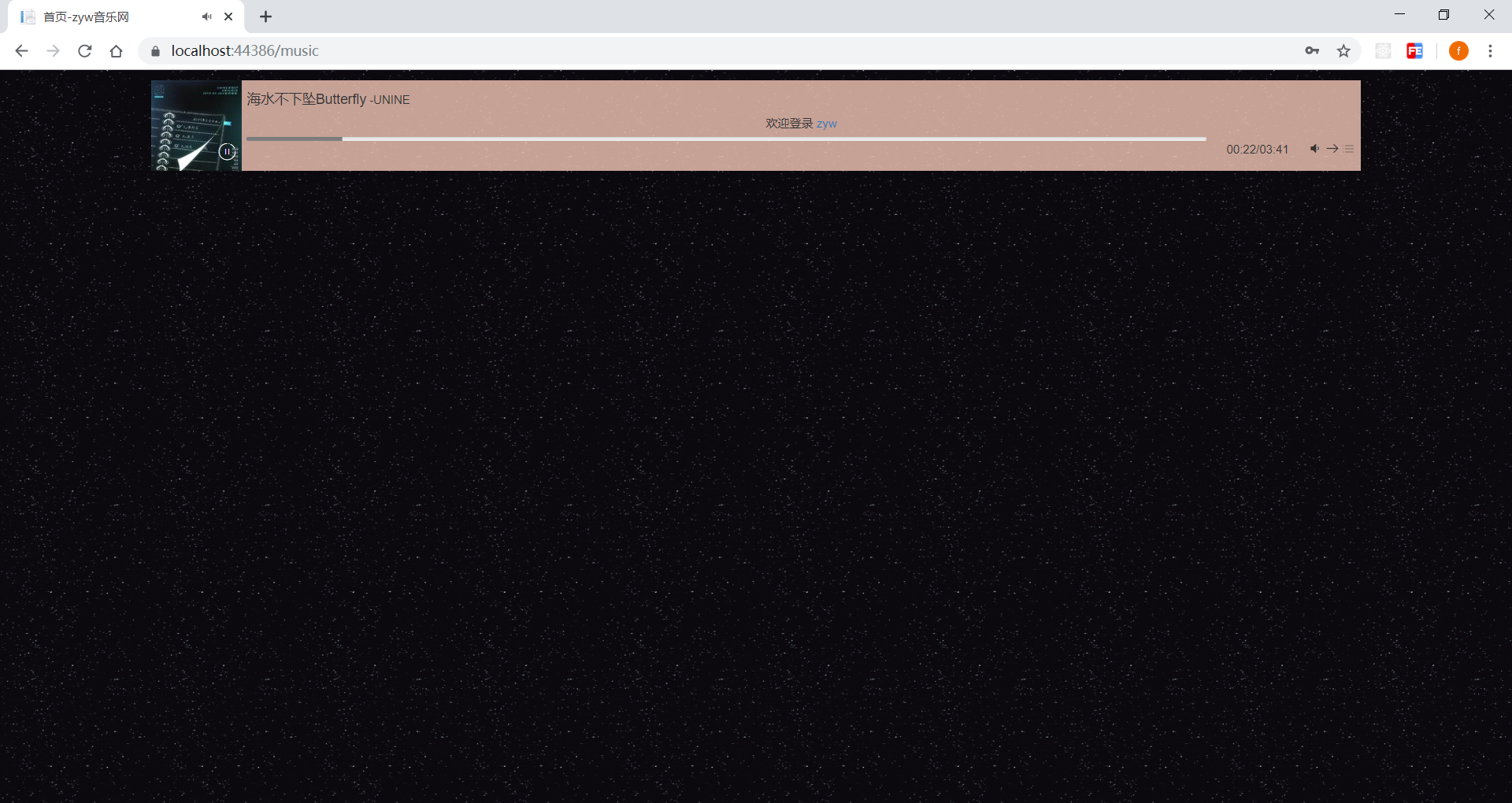


图 2 主页

**实验环境及实验工具**

本次实验实验环境是：电脑（Windows+.Net）+VS；

用到的工具：Photoshop、Adobe Dreamweaver、VisualStudio；

本次实验用到了H5标签audio，充分利用了jquery框架与bootstrap框架。后端在.net环境下书写c#完成数据库的动态操作，从而实现动态网站开发。

**参考文献**

1. 江广顺, 余松编著. asp.net 2.0+sql server热门网络应用开发详解. 北京：人民邮电出版社，2007
2. 吴晨,张亮,张静编著. asp.net + sql server数据库开发与实例. 北京：清华大学出版社，2006
3. 赵爱美.基于HTML5和\_NET的移动学习平台研究与实现[J].河南科技学院学报，2013（8）：62-66.