中南大學
CENTRAL SOUTH UNIVERSITY

WEB 技术
实验报告

学生姓名	
学　　号	
专业班级	
指导教师	
学　　院	计算机学院
完成时间	

目 录

一　实践任务描述

(1) 实现前端页面的基本布局。要求：

- ✧ 布局类似于学校门户 http://my.csu.edu.cn/login/index.jsp
- ✧ 顶部需有 LOGO 栏目；
- ✧ 左侧提供轮播图；
- ✧ 提供账号密码登录方式；
- ✧ 提供手机号码（邮箱）+验证码登录方式；
- ✧ 登录成功后跳到显示“登录成功”四字的页面（简单设计）；
- ✧ 提供忘记密码和修改密码功能；
- ✧ 提供用户的增删改查。

(2) 完成前后端数据交互（用 JSON 格式）

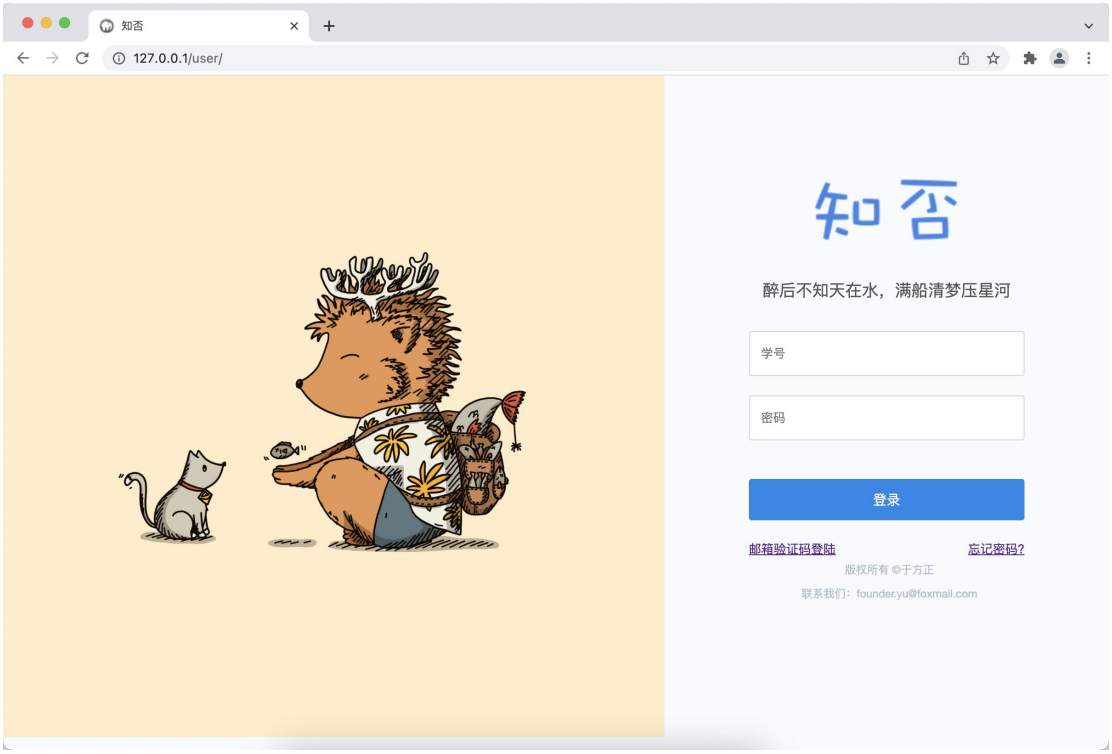
(3) 数据操作要求：

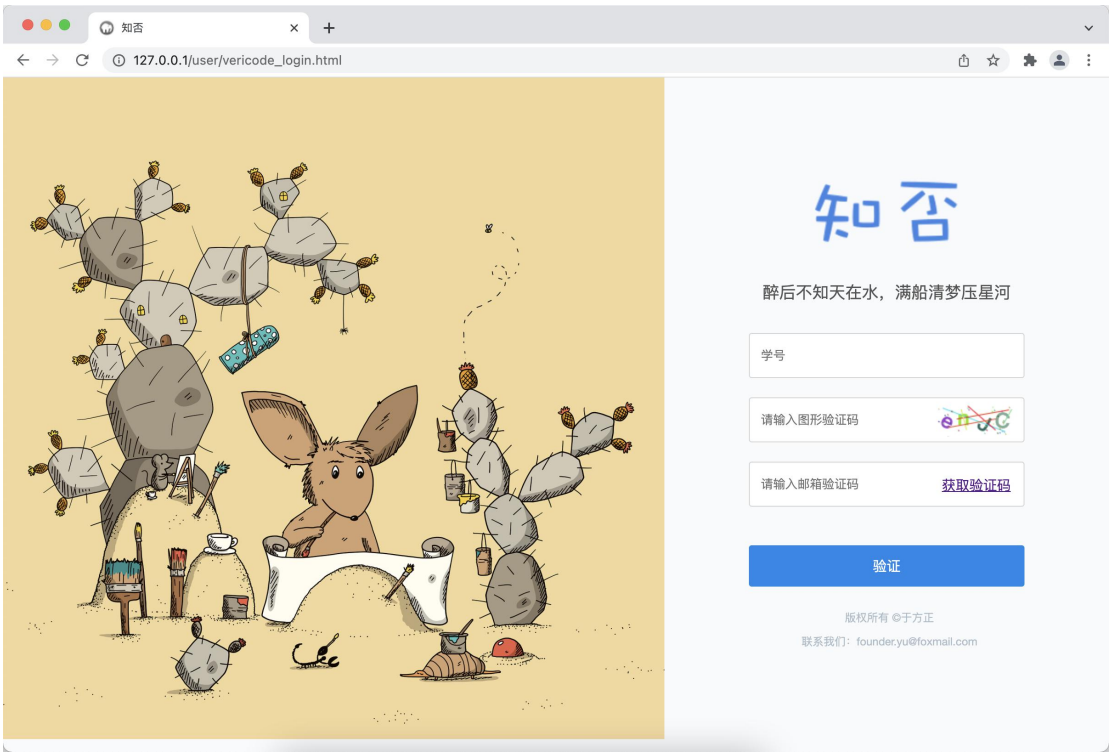
- ✧ 数据统一存储在后端数据库中；
- ✧ 账号密码登录方式需进行验证，验证通过方能登录；
- ✧ 手机（邮箱）验证码需调用第三方短信接口发送验证码并进行验证；
- ✧ 后端实现技术不限、数据库系统不限。

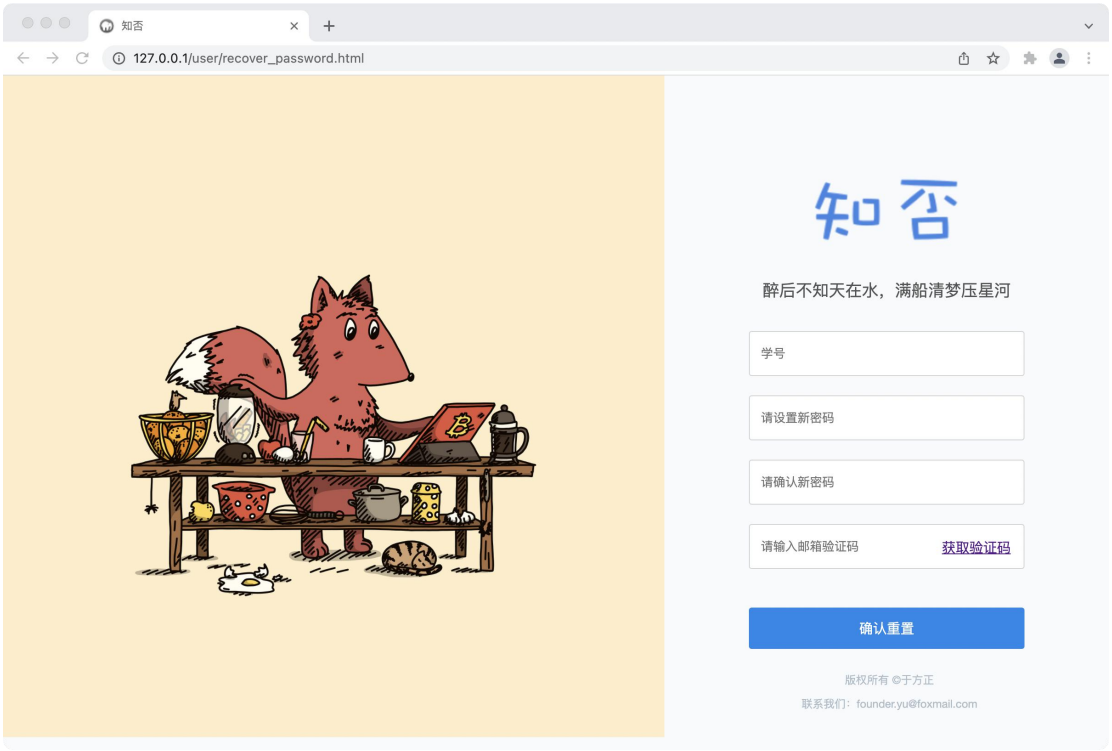
二　项目文件结构及功能

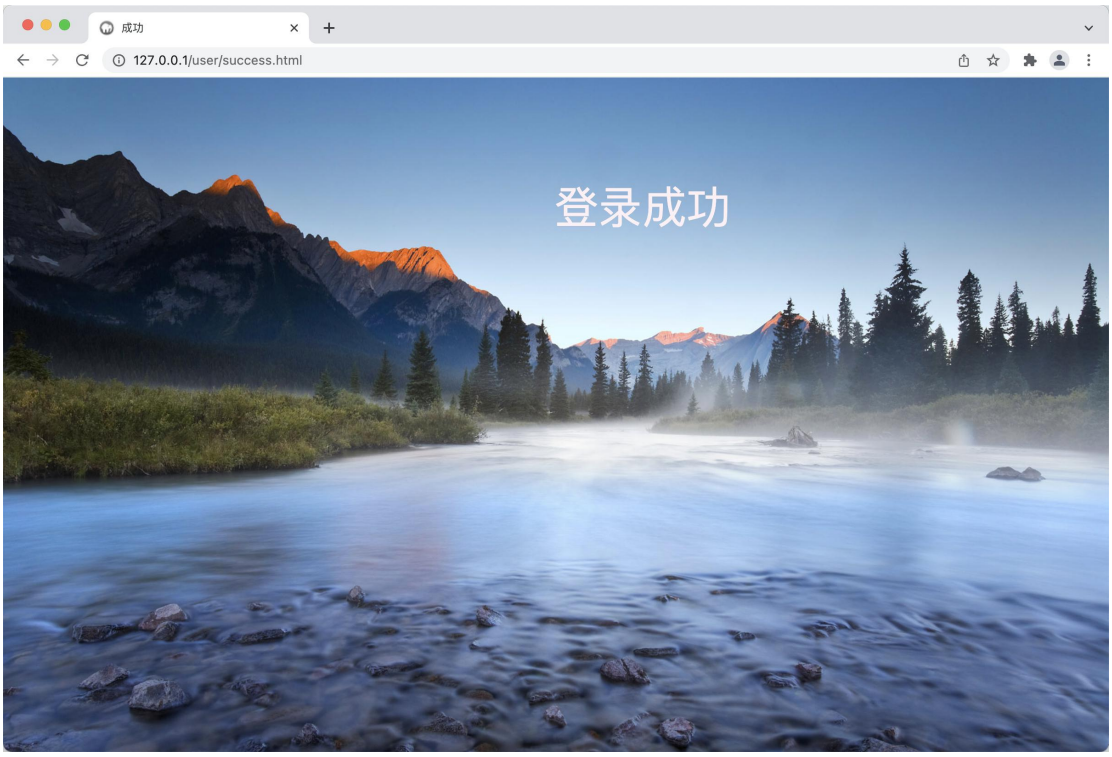
/user	index.html	用户首页
	recover_password.html	恢复密码页面
	vericode_login.html	验证码登录页面
	success.html	登录成功页面
	login.php	处理主页提交的 form，验证用户名密码
	recover.php	重置用户密码为用户设定的新密码
	send.php	发送邮箱验证码
	verify.php	验证邮箱验证码
/admin	index.php	管理员主页
	add.php	增加用户
	del.php	删除用户
	edit.php	编辑用户信息的提示页面
	edit_post.php	将修改过的用户信息应用到数据库
	reset.php	重置用户密码
/js	carousel.js	轮播图 js 实现
/css	base.css	基础样式
	bootstrap.css	管理员主页样式
/img	1.png	轮播图 1
	2.png	轮播图 2
	3.png	轮播图 3
	BG1.jpg	登录成功页背景图
	zhifou_logo.png	网页 LOGO

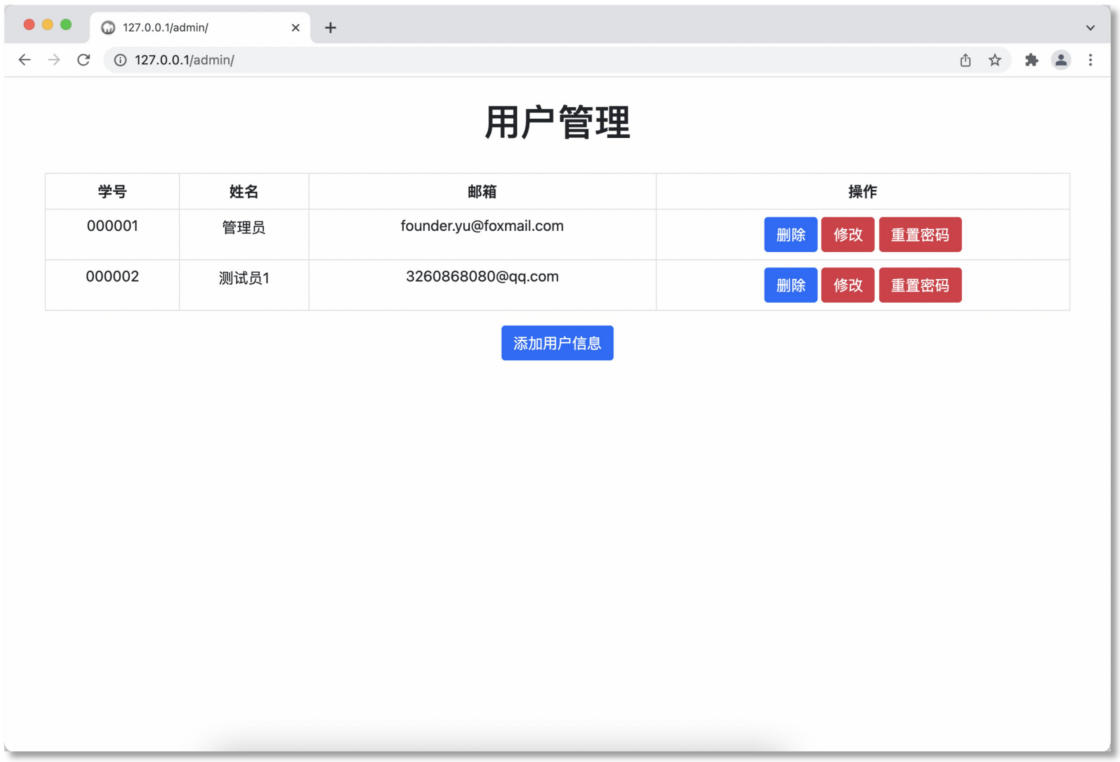
三　页面效果展示

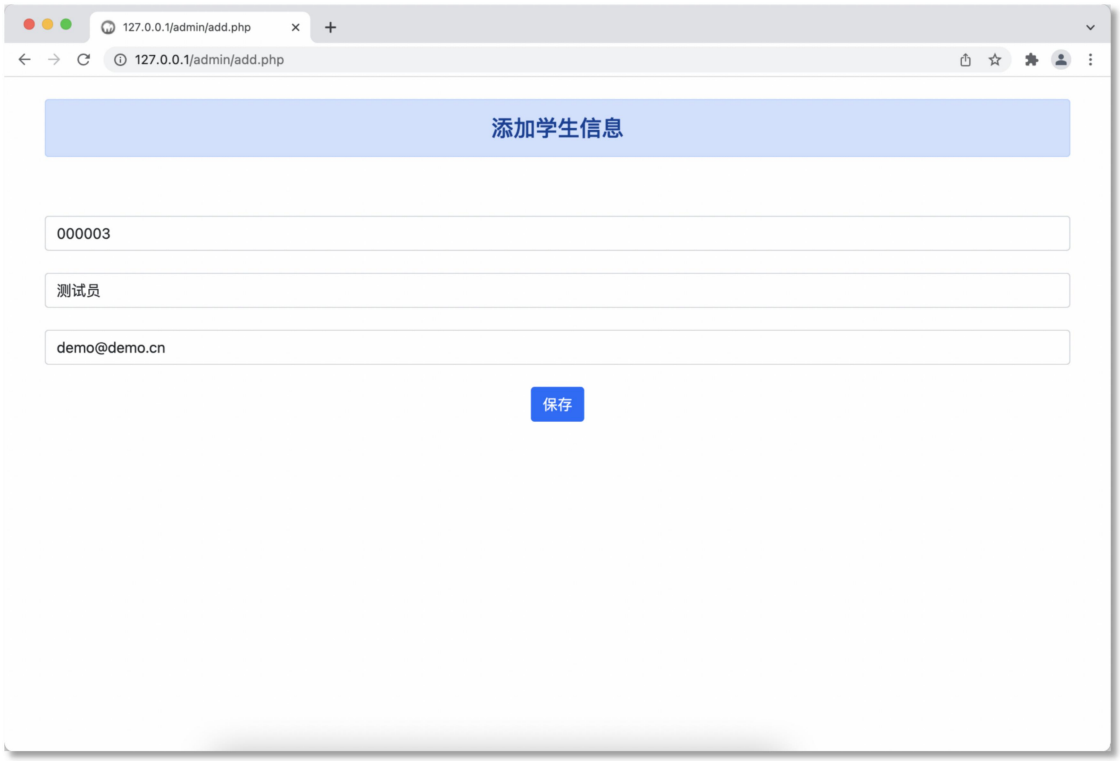

知否
127.0.0.1/user/
知否
醉后不知天在水，满船清梦压星河
学号
密码
登录
邮箱验证码登陆
忘记密码?
版权所有 ©于方正
联系我们：founder.yu@foxmail.com


知否
127.0.0.1/user/vericode_login.html
知否
醉后不知天在水，满船清梦压星河
学号
请输入图形验证码
请输入邮箱验证码
获取验证码
验证
版权所有 ©于方正
联系我们：founder.yu@foxmail.com


知否
127.0.0.1/user/recover_password.html
知否
醉后不知天在水，满船清梦压星河
学号
请设置新密码
请确认新密码
请输入邮箱验证码
获取验证码
确认重置
版权所有 ©于方正
联系我们：founder.yu@foxmail.com

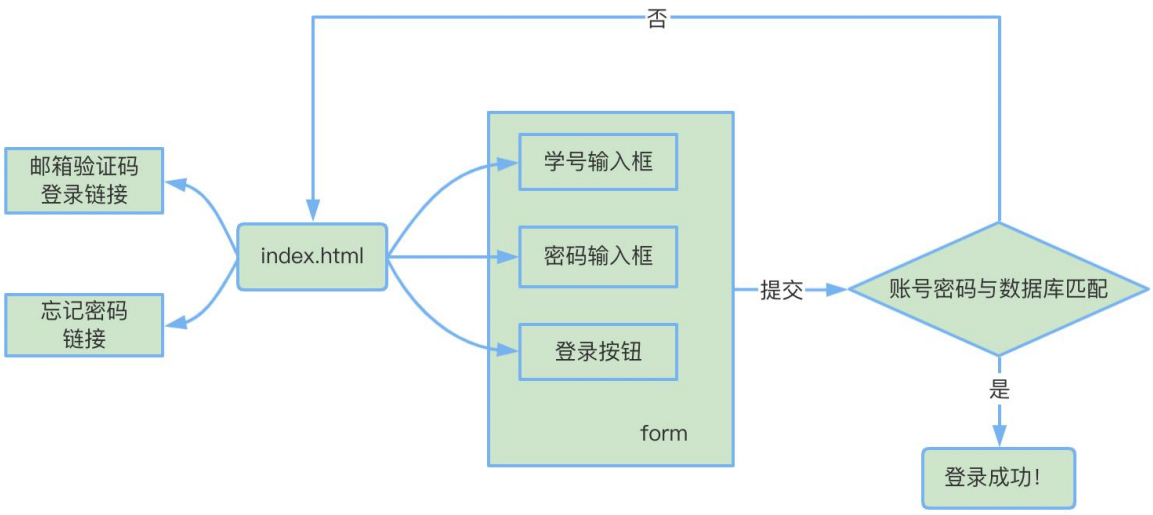

成功
127.0.0.1/user/success.html
登录成功


127.0.0.1/admin/
127.0.0.1/admin/
用户管理
学号
姓名
邮箱
操作
000001
管理员
founder.yu@foxmail.com
删除
修改
重置密码
000002
测试员1
3260868080@qq.com
删除
修改
重置密码
添加用户信息

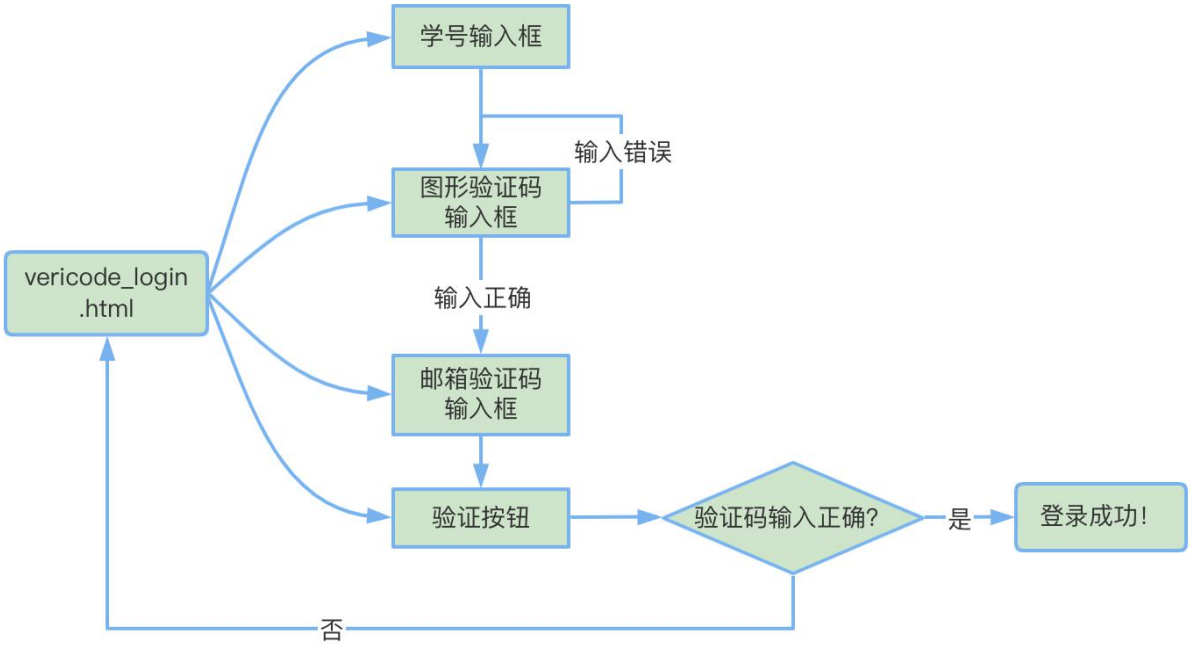

127.0.0.1/admin/add.php
127.0.0.1/admin/add.php
添加学生信息
000003
测试员
demo@demo.cn
保存

四　页面逻辑图

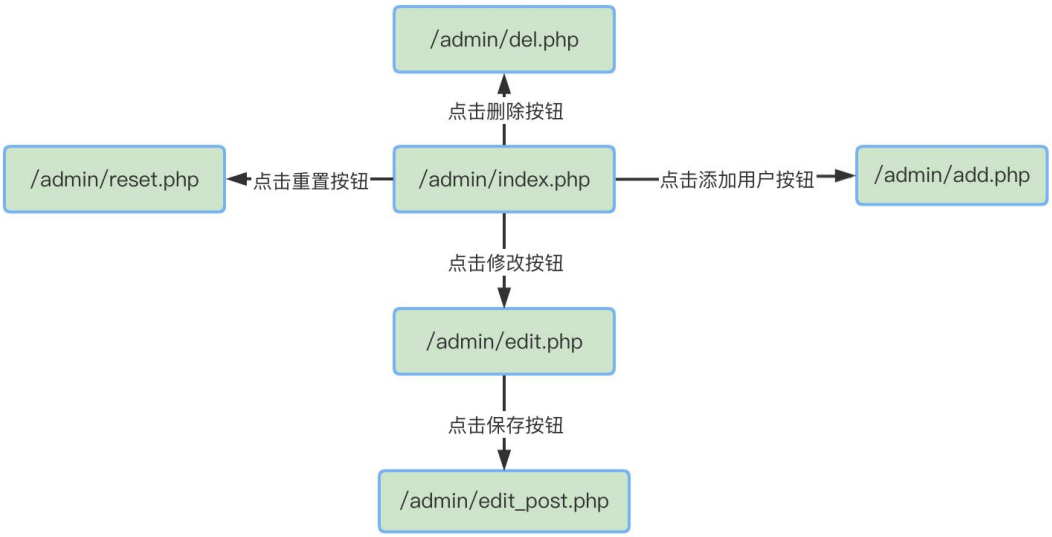
4.1　index.html



4.2　vericode_login.html



4.3 /admin


/admin/del.php
点击删除按钮
/admin/reset.php
点击重置按钮
/admin/index.php
点击添加用户按钮
/admin/add.php
点击修改按钮
/admin/edit.php
点击保存按钮
/admin/edit_post.php

五　核心技术点

5.1　CSS 选择器及优先级

5.1.1　CSS 选择器

选择器		eg	描述
通用选择器	*	*	选择所有元素。
类选择器	.class	.message	选择 class="intro" 的所有元素。
id 选择器	#id	#head	选择 id="firstname" 的所有元素。
元素选择器	el	p	选择所有 <p> 元素。
选择器分组	el,el	div,p	选择所有 元素和所有 <p> 元素。
后代选择器	el el	div p	选择 元素内部的所有 <p> 元素。
子元素选择器	el > el	div>p	选择 的第一子代的 所有<p> 元素。
相邻兄弟选择器	el + el	div+p	选择与 同级且紧接在其后的第一个 <p> 元素

5.1.2　CSS 选择器优先级序列

- ✧ 在属性后面使用 !important 会覆盖页面内任何位置定义的元素样式。
- ✧ 作为 style 属性写在元素内的样式
- ✧ id 选择器
- ✧ 类选择器
- ✧ 元素选择器
- ✧ 通配符选择器
- ✧ 浏览器自定义或继承

5.2　邮箱验证码发送

(1) Outlook 邮箱设置：登录 outlook 邮箱 → 设置 → 邮件 → 同步电子邮件记录 SMTP 服务器信息、端口号及加密方式。

(2) 下载 PHPMailer：PHPMailer 是一个用于发送电子邮件的 PHP 函数包。

(3) 利用 PHPMailer 发送邮件

```
$mail = new PHPMailer(true);          // Passing `true` enables exceptionstry {
    //服务器配置
    $mail->CharSet ="UTF-8";                                    //设定邮件编码
    $mail->SMTPDebug = 0;                                       // 调试模式输出
    $mail->isSMTP();                                            // 使用 SMTP
    $mail->Host = 'smtp.office365.com';                 // SMTP 服务器
    $mail->SMTPAuth = true;                                     // 允许 SMTP 认证
    $mail->Username = 'f******u@outlook.com';   // SMTP 用户名 即邮箱的用户名
    $mail->Password = '**********';                             // SMTP 密码  部分邮箱是授权码
    $mail->SMTPSecure = 'STARTTLS';                 // 允许 TLS 或者 ssl 协议
    $mail->Port = 587;                                          // 服务器端口
    $mail->setFrom('f*****u@outlook.com', 'ZhiFou');  //发件人（以 QQ 邮箱为例）
    $mail->addAddress($email, $name);           // 收件人
    $mail->addReplyTo('founder.yu@outlook.com', 'info'); //回复的时候回复给哪个邮箱 建议和发件人一致
    $verify = codestr();                                        //此处为调用随机验证码函数
    $_SESSION[$id]=$verify;
    $mail->isHTML(true);                                                // 是否以 HTML 文档格式发送  发送后客户端可直接显示对应 HTML 内容
    $mail->Subject = '「知否」身份登录验证';
```

```
    $mail->Body    = '<h1>欢迎使用【知否】</h1><h3>您的身份验证码是：<span>'.$verify.'</span>，如果您没有提交登录请求，则可放心忽略此邮件。</h3>' . date('Y-m-d H:i:s');
    $mail->AltBody = '欢迎使用 知否,您的身份验证码是：'.$verify .' 如果您没有提交登录请求，则可放心忽略此邮件。 '.date('Y-m-d H:i:s');
    $mail->send();
    echo '验证邮件发送成功，请注意查收！ ';
} catch (Exception $e) {
    echo '邮件发送失败: ', $mail->ErrorInfo;
}
```

5.3 图形验证码绘画

图形验证码的实现选择在 canvas 上绘画大写字母、小写字母、数字，每个元素随机偏转 0-30 度，之后再绘制五根颜色随机、位置随机的线条以及 30 个颜色随机、位置随机的污点。具体代码实现如下：

```
function draw(show_num) {
    var canvas_width=document.getElementById('canvas').clientWidth;
    var canvas_height=document.getElementById('canvas').clientHeight;
    var canvas = document.getElementById("canvas");//获取到 canvas 的对象，演员
    var context = canvas.getContext("2d");//获取到 canvas 画图的环境，演员表演的舞台
    canvas.width = canvas_width;
    canvas.height = canvas_height;
    var sCode =
"A,B,C,E,F,G,H,J,K,L,M,N,P,Q,R,S,T,W,X,Y,Z,1,2,3,4,5,6,7,8,9,0,q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z,x,c,v,b,n,m";
    var aCode = sCode.split(",");
    var aLength = aCode.length;//获取到数组的长度

    for (var i = 0; i <= 3; i++) {
        var j = Math.floor(Math.random() * aLength);//获取到随机的索引值
        var deg = Math.random() * 30 * Math.PI / 180;//产生 0~30 之间的随机弧度
        var txt = aCode[j];//得到随机的一个内容
        show_num[i] = txt;
        var x = 2+i * 20;//文字在 canvas 上的 x 坐标
        var y = 20 + Math.random() * 8;//文字在 canvas 上的 y 坐标
```

```
            context.font = "bold 23px 微软雅黑";

            context.translate(x, y);
            context.rotate(deg);

            context.fillStyle = randomColor();
            context.fillText(txt, 0, 0);

            context.rotate(-deg);
            context.translate(-x, -y);
        }
        for (var i = 0; i <= 5; i++) { //验证码上显示线条
            context.strokeStyle = randomColor();
            context.beginPath();
            context.moveTo(Math.random() * canvas_width, Math.random() * canvas_height);
            context.lineTo(Math.random() * canvas_width, Math.random() * canvas_height);
            context.stroke();
        }
        for (var i = 0; i <= 30; i++) { //验证码上显示小点
            context.strokeStyle = randomColor();
            context.beginPath();
            var x = Math.random() * canvas_width;
            var y = Math.random() * canvas_height;
            context.moveTo(x, y);
            context.lineTo(x + 1, y + 1);
            context.stroke();
        }
    }
```

5.4 轮播图 JS 实现

通过多张图片平铺，用 overflow:hidden 只显示一张图片、其他的隐藏，无缝滚动用定时器改变元素的 left 值让图片呈现左右滚动的效果。

容器 #container 中包含#lis，#lis 里内有多个<a>标签，#lis 比#container 宽，通过计算偏移量利用定时器实现自动播放，或通过手动点击事件切换图片。图中红色为#container 显示部分 黑色为#lis 隐藏部分。具体 Javascript 代码实现如下：

```
var imgArr = []; // 图片数组 var curIndex = 0; // 当前显示图片 var timer = null; // 定时器

function slide(slideContainer){

    var width = imgArr[curIndex].width; // 获取图片宽度，也就是每次切换图片要滑动的距离

    var distanceMoved = 0; // 已经移动的距离

    var step = 10; //切换的步长

    var curConLeft = slideContainer.offsetLeft; // 获取 ul 的 left

    var slideInterval = setInterval(function (){ // 此定时器是为了实现切换动画

        if(Math.abs(width – distanceMoved) > step){ // 边界判定，判断已移动距离以及应移动距离的差与步长关系

            curConLeft –= step; // 大于步长则不断移动

            slideContainer.style.left = `${curConLeft}px`; // 移动

            distanceMoved += step; // 已移动距离加步长

        }else{

            clearInterval(slideInterval); // 若最后移动距离不足步长，则清除动画定时器

            slideContainer.style.left = `${curConLeft – width + distanceMoved }px`; // 直接完成此次动画

            distanceMoved = 0; // 重设移动距离为 0
```

```
            if(++curIndex === imgArr.length){ // index 加 1，判断是否为最后一张来作边缘处理
                curIndex = 0; // 最后一张则重置 index
                slideContainer.style.left = 0;  // 重置 ul
            }
        }
    }, 10);
}

(function start() {
    var config = {
        height: "720px", // 配置高度
        interval: 5000 // 配置轮播时间间隔
    }
    document.getElementById("carousel").style.height = config.height; // 将轮播容器高度设定
    document.querySelectorAll("#carousel img").forEach(v => imgArr.push(v)); // 获取所有图片组成数组
    var slideContainer = document.querySelector("#carousel > ul"); // 获取 ul 也就是一行图片的容器
    var li = document.createElement("li"); // 创建<li>
    var img = document.createElement("img"); // 创建新的<img>
    img.src = imgArr[0].src; // 设置图片 src
    li.appendChild(img); // 追加<img>到<li>
    slideContainer.appendChild(li); // 将第一张图片追加到轮播图的最后，作边缘处理
    timer = setInterval(() => {slide(slideContainer)},config.interval); // 设置定时器定时切换
```

5.5 PHP 数据库操作

(1) 链接数据库

mysqli_connect('主机地址','mysql 用户名','mysql 密码 ','数据库','端口号')

返回：如果连接成功，返回资源类型的标志符号；如果连接失败，返回 false。如果我们与 mysql 建立的连接不只一条，那么以后操作数据库的各种函数都必须传入返回的连接符号；如果我们与 mysql 建立的连接只有一条，那么以后操作数据库的各种函数就不必传入这个标识符号。建议都传入。密码为空可以省略密码。

(2) 检测数据库连接是否成功

mysqli_connect_errno()与 mysqli_connect_error()

mysqli_connect_errno(); 返回上次连接数据库错误的错误号，连接成功返回 0

mysqli_connect_error(); 返回上次连接数据库的错误信息

```
if(mysqli_connect_errno($conn)){
    die("数据库连接失败！失败信息：".mysqli_connect_error($conn));
}
```

前面两步合并的写法：连接数据库同时判断

```
$conn = mysqli_connect("localhost", "root", "", "mydb") or die("数据库连接失败！失败信息：
".mysqli_connect_error($conn));
```

(3) 选择数据库

mysqli_select_db($link,$dbname)

参数①：标识符

参数②：连接数据库名称

连接成功，返回 true；连接失败，返回 false。如果修改数据库成功，则资源标识符中的数据库就会发生变动;如果修改失败而没有通过代码终止操作，则后续代码可以使用原数据库继续执行 mysqli_select_db($conn, "mydb") or die("数据库选择失败！ ");

(4) 设置字符集编码格式

mysqli_set_charset($link,$charset) 只能设置为 utf8 而不能是 utf-8

mysqli_set_charset($conn,"utf8") or die("数据库编码集设置失败！ ");

(5) 编写 sql 语句

$sql = "select * from 表名";

(6) 执行 sql 语句

mysqli_query($link,$sql)

如果是（DML）增、删、改，将返回布尔类型是否成功返回上一次操作时受影响的行数；如果是（DQL）查询，将返回资源结果集；如果查询失败，返回 false；

(7) 解析结果集

var_dump(mysqli_fetch_array($res)); 处理结果集，返回关联数组和索引数组

参数①：需要处理的结果集

参数②：返回哪种数组格式

MYSQL_ASSOC - 关联数组

MYSQL_NUM - 数字数组

MYSQL_BOTH - 默认。同时产生关联和数字数组

(8) 关闭资源与结果集

mysqli_free_result()和 mysqli_close()

mysqli_free_result($res); //释放查询资源结果集

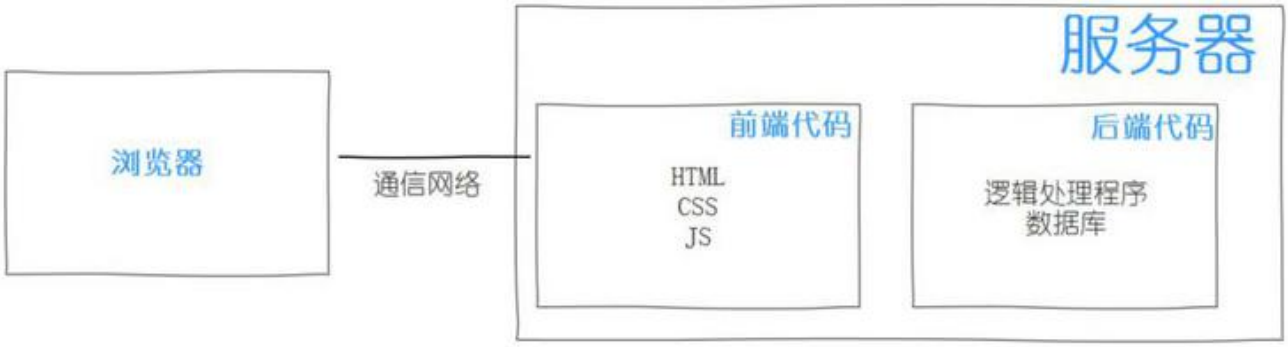
mysqli_close($conn); //关闭数据库连接

5.6 前后端数据交互

5.6.1 前端与后端

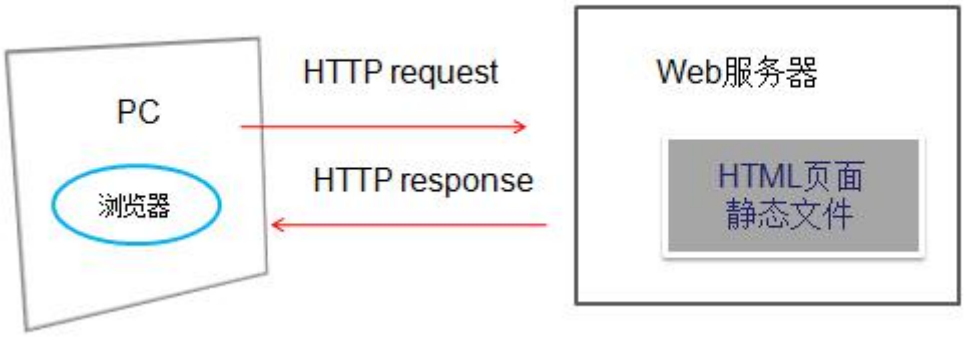
浏览器显示的网页即为 web 前端界面，提供用户与网站进行交互的可视化接口，而 web 后端服务主要指在服务器中执行的逻辑运算和数据处理，它为前端提供着访问服务。所谓的前后端只是从代码被执行的位置来区分的，前端代码在用户面前被执行，后端代码在遥远的服务

器上被执行。但是，无论前端或后端代码，都是存放在服务器上的，只是当浏览器请求的时候，从服务器发送过去而已。



5.6.2 AJAX 介绍

在上述 web 应用工作的原理中，我们通过 HTTP 协议访问一个在服务端存在的文件，服务器可以找到该文件并将其内容封装到 HTTP 请求中，以消息体的形式返回给客户端。不过这种方式只能访问静态的页面，无法与后端数据库进行交互。既然用户需要通过 web 前端实时与后端数据库进行交互，那么网页也需要动态的更新，如果每次更新一个数据都通过重新获取 Html 文件的方式来实现势必会导致网络负荷加重，页面加载迟缓。而 Ajax 技术可以很好的解决这个问题。



Ajax 即异步 JavaScript 和 XML，是一种创建交互式网页的技术，可以不重新加载整个网页的情况下更新部分网页。目前 jQuery 库提供多个与 AJAX 相关的方法。通过 jQuery AJAX 方法，能够使用 HTTP Get 和 HTTP Post 从远程服务器上请求文本、HTML、XML 或 JSON，同时能够把这些外部数据直接载入网页被选元素中。

作为 web 开发人员广泛采用的 JavaScript 封装库之一的 jQuery 库，它可以极大地简化我们

的 JavaScript 编程，缓解浏览器之间不兼容的影响，要知道在不同浏览器中进行 web 网页的兼容性测试也是一个不小的工作量。我们可以通过一个简单的例子发现 jQuery 库的优势所在：

```
$("p.neat").addClass("ohmy").show("slow");
```

通过以上简短的代码，开发者可以遍历“neat”类中所有的<p>元素，然后向其增加“ohmy”类，同时以动画效果缓缓显示每一个段落。开发者无需检查客户端浏览器类型，无需编写循环代码，无需编写复杂的动画函数，仅仅通过一行代码就能实现上述效果。jQuery 的口号“最少的代码做最多的事情”果真名副其实，它把 JavaScript 带到了一个更高的层次。

5.6.3 JSON 格式介绍

对于交互的数据格式，这里采用 JSON(JavaScript Object Notation)，它是一种轻量级的数据交换格式，采用完全独立于编程语言的文本格式来存储和表示数据。JSON 键值的层次结构简洁清晰，易于阅读和编写，使得 JSON 成为理想的数据交换语言。举个例子来理解 JSON 数据格式：

```
{//JSON 键/值对
 “'wJsona'” :” kkk”
 “'wjsonb'” :” 12”
 “'wjsonc” :” 80”
}
```

5.6.4 代码实现

下面介绍前端 jQuery .ajax()请求 JSON 数据的方法，代码如下：

```
    function useTestFun() {
        $.ajax({
            url: "/Usedefine",//获取数据的 URL
            data:JSON.stringify({
                'wJsona':"kkk",
                'wjsonb':12,
                'wjsonc':80,
            }),
            type: "POST",//HTTP 请求方法
```

```
            dataType:'JSON',//获取数据执行方式
            success:function(data){
                if(data.status == 'True'){//传入为 JSON 对象格式
                    alert('连接成功');
                }
                else{
                    $("#labletip").show();
                }
            },
            error:function(err){
                alert('连接失败');
            }
        });
    }
```

在数据传输过程中，JSON 是以文本，即字符串的形式传递的，而 JS 操作的是 JSON 对象，所以 JSON 对象和 JSON 字符串之间的相互转换是关键，可以使用 JSON.stringify()将 JSON 对象转化为 JSON 字符串，使用 JSON.parse()将 JSON 字符串转换为 JSON 对象。

```
JSON 字符串:
var str1 = '{ "name": "cxh", "sex": "man" }';
JSON 对象:
var str2 = { "name": "cxh", "sex": "man" };
var obj = str.parseJSON(); //由 JSON 字符串转换为 JSON 对象
var obj = JSON.parse(str); //由 JSON 字符串转换为 JSON 对象
var last = obj.toJSONString(); //将 JSON 对象转化为 JSON 字符
var last = JSON.stringify(obj); //将 JSON 对象转化为 JSON 字符
```

六　心得与体会

本学期学习，老师从网页基础、HTML 入手，攫取重点，给我们介绍了它的相关知识。我们主要学习了 HTML5，里面有很多的新特性且时下较为流行。它相当于一个网页界面的宏观架构。如果把一个网页的实现比作是一座建筑的建造过程，那么 HTML 即是这座建筑里的钢筋混泥土，搭建起整个建筑的框架、承重。

之后，我们又学习了 CSS 基础样式，仍然借用上面的比喻，CSS 则相当于建筑里各个房间的不同结构，它们使得这座建筑更加的多样化。且相对于 HTML，它更加的复杂、多样化，呈现的效果也具有更多的可能性。老师向我们推荐了《CSS 禅意花园》这本书，里面列举了丰富且多种多样的 CSS 样式。

作为今后可能成为的优秀前端工程师，仅仅学习自己内部的知识是远远不够的。因此，第一个月中我们也学习了 UI 中 PS 绘图制作基础，这对于一个前端来说也是非常重要的，在今后的工作中，我们可能会面对各种各样的问题，如果掌握部分 PS 尤其是切图技术，在和 UI 的接触中可以减少很多不必要的繁琐工作。

总之，作为一个前端工程师，我们所要掌握的知识是全面的，当我们写代码时的思维是缜密的。HTML 和 CSS 是基础中的基础。之后我们会学习更多的 JavaScript 相关知识和其他，希望自己在这过程中仍能保持谦逊的的心态，去学习前人留下的珍贵宝藏。

参考资料

[1] 张工厂.PHP7+MySQL8 动态网站开发[M].北京：清华大学出版社,2020.

[2] 孙俏,祖明,王新阳.Web 前端开发[M].北京：高等教育出版社,2021.