**数据库课程设计**

题 目 党的基本知识学习测试系统

专业班级 计算2101

姓 名 孙国铭

学 号 2107010118

# 需求分析

## 数据结构

用户信息：包括用户名id、密码、邮箱、角色（管理员、组织员、入党积极分子）

题库信息：包括题目ID、题目内容、选项、答案

考试信息：包括试卷号、用户名、成绩、做题时间

试卷信息记录：包括试卷号、题号、试卷开始时间、结束时间

学习记录：包括用户名、成绩、开始时间、结束时间

统计记录：包括登录次数、学习次数、时间

表1.(用户)数据结构表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| 用户名 | 唯一标识每位用户 | Varchar | 20 |
| 登录密码 | 用户登录密码 | Varchar | 20 |
| 邮箱 | 用户个人邮箱 | Varchar | 20 |
| 角色 | 用于身份登录 | Varchar | 20 |

表2.(题库)数据结构表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| 题目编号 | 唯一标识题号 | Varchar | 20 |
| 题目内容 | 题目内容 | Varchar | 200 |
| 选项A | 选项A | Varchar | 20 |
| 选项B | 选项B | Varchar | 20 |
| 选项C | 选项C | Varchar | 20 |
| 选项D | 选项D | Varchar | 20 |
| 答案 | 题目答案 | Varchar | 20 |

表3.(自学自考记录)数据结构表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| 用户名 | 唯一标识用户 | Varchar | 20 |
| 成绩 | 自考成绩 | Varchar | 20 |
| 开始时间 | 做题开始时间 | Varchar | 20 |
| 结束时间 | 做题结束时间 | Varchar | 20 |

表4.(试卷)数据结构表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| 试卷编号 | 试卷编号 | Varchar | 20 |
| 题号 | 试卷中有哪些题目号 | Varchar | 20 |
| 开始时间 | 试卷做题开始时间 | Varchar | 20 |
| 结束时间 | 结束时间 | Varchar | 20 |

表5.(考察记录)数据结构表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| 试卷编号 | 试卷编号，唯一标识 | Varchar | 20 |
| 用户名 | 用户名，唯一标识 | Varchar | 20 |
| 成绩 | 试卷成绩 | Varchar | 20 |
| 做题时间 | 做题花费时间 | Varchar | 20 |

表6.(登录/学习记录)数据结构表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| 登录次数 | 唯一标识 | Int | 11 |
| 学习次数 | 唯一标识 | Int | 11 |
| 时间 | 时间 | Varchar | 20 |

## 数据流

1. 用户登录流程：用户打开系统页面，输入用户名和密码进行登录，系统验证用户身份，根据用户的角色和权限控制用户的操作。

2. 题库维护流程：管理员或组织员登录系统，选择题库维护功能，添加、修改、删除题目信息，同时可以对题目进行分类、标记、搜索等操作，将题目保存到题库中。

3. 学员学习流程：入党积极分子登录系统，选择学习功能，系统随机生成试卷，学员选择答案并提交试卷，系统自动评分并显示答题情况和正确答案，同时记录学习记录和统计信息。

4. 考察流程：组织员登录系统，选择考察功能，选择考试题目和考试时间，将考试信息发布给考察对象，考察对象登录系统，进行考试并提交答案，系统自动评分并显示答题情况和正确答案，同时记录考试结果和统计信息。

5.统计展示流程：学员所做的试卷进行总的统计，用列表和柱状图的形式直观的看出试卷的效果等等

表7.数据流表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据流名 | 数据流来源 | 数据流去向 | 组成 |
| 登录状况 | 登录 | 登入详情 | 数据结构 |
| 题库增删 | 提交申请 | 数据库 | 数据结构 |
| 模式选择 | 用户点击 | 模式跳转 | 数据结构 |
| 考核流程 | 信息设置 | 数据库 | 数据结构 |
| 统计流程 | 数据库 | 考试详情 | 数据结构 |

## 数据存储

1. 用户信息存储：需要存储用户的基本信息，包括用户名、密码、角色、权限等。

2. 题库信息存储：需要存储题目的基本信息，包括题目编号、题目类型、题目内容、答案选项、正确答案等。

3. 学员学习记录存储：需要存储学员的学习记录，包括学员的答题情况、答题时间、得分情况等。

4. 考察记录存储：需要存储考察记录，包括考察对象的姓名、考试时间、考试成绩等。

5. 试卷记录存储：包括生成的试卷号和试卷包含的题号等信息。

6. 统计记录存储：包括登录/学习次数等信息。

表8.数据存储表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据存储名 | 输入的数据流 | 输出的数据流 | 组成 |
| 用户表 | 已注册的用户 |  | 数据结构 |
| 题库表 | 已有题库 |  | 数据结构 |
| 学习记录表 | 已达试题详情 |  | 数据结构 |
| 考察表 | 测试详情 |  | 数据结构 |
| 试卷信息表 | 试题组成 |  | 数据结构 |
| 统计表 | 登录/学习提交 |  | 数据结构 |

## 处理过程

1. 用户认证和授权：用户在使用系统之前需要进行认证和授权，系统需要对用户的身份进行验证，并根据用户的角色和权限对其进行授权，以保证只有具备相应权限的用户才能进行相关操作。

2. 试题生成和展示：系统需要从题库中随机选取一定数量的试题，并将其展示给学员进行学习和测试。试题的数量和类型可以根据学员的学习进度和要求进行调整。

3. 答题和批改：学员需要在系统中作答，并提交答案。系统需要对答案进行自动批改，并给出得分和评价。针对主观题，系统需要安排人工批改，确保答案的准确性和公正性。

4. 学习、考察记录和统计：系统需要记录学员的学习记录，包括学习时长、答题情况、得分情况等，以便学员进行自我评估和系统进行统计分析。同时，系统需要对各种统计信息进行计算和分析，例如学习人数、学习效果、谈话考核情况等，以便组织部门进行管理和决策。

# 概念结构设计

## E-R图(实体、属性、联系)

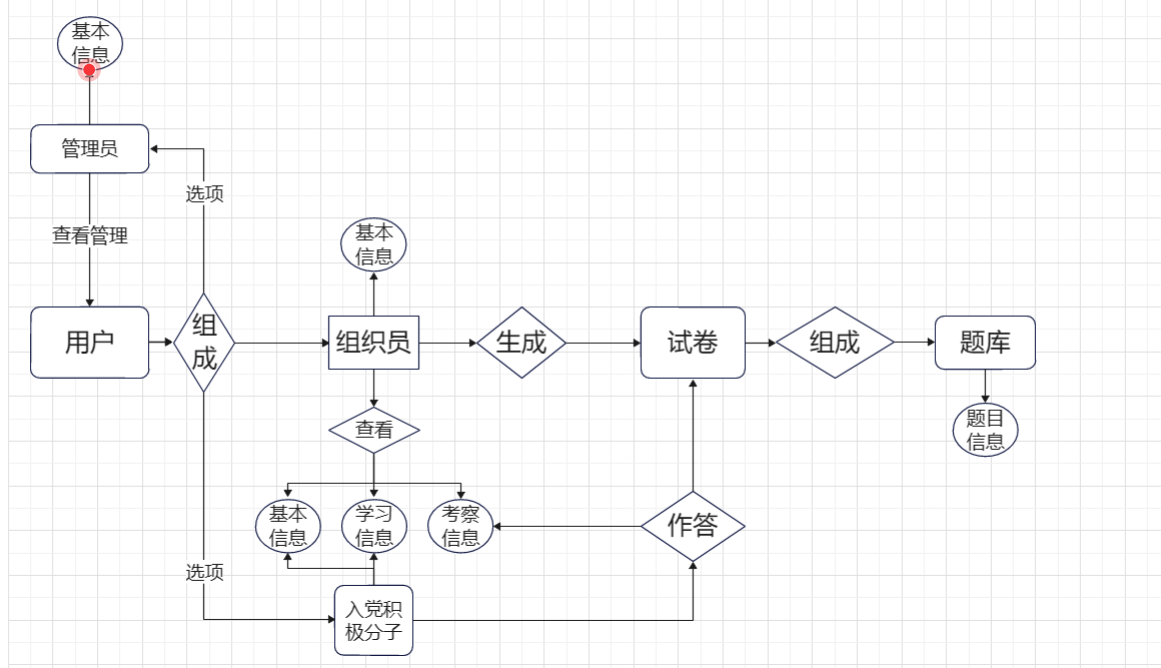


图1.系统的基本E-R图

1. 用户实体：包括管理员、组织员和入党积极分子等，需要存储用户的基本信息和权限等信息。

2. 题目实体：需要存储题目的内容、选项、答案和解析等信息。

3. 试卷实体：包括生成的试卷和学员提交的答卷等，需要存储试卷的相关信息和答卷的答案等信息。

4. 学习记录实体：包括学员的学习记录和系统的统计信息等，需要存储学习的时间、内容和效果等信息。

4. 考察记录实体：包括学员的考察记录和系统的统计信息等，需要存储试卷做题的时间、内容和效果等信息。

## 系统说明书

## 系统要求

系统的功能和特性，主要包括用户注册登录、用户管理、试题库管理、试卷生成、答卷评分、学习记录统计、考察记录统计七大部分。

## 数据流图

**(1)数据流图**

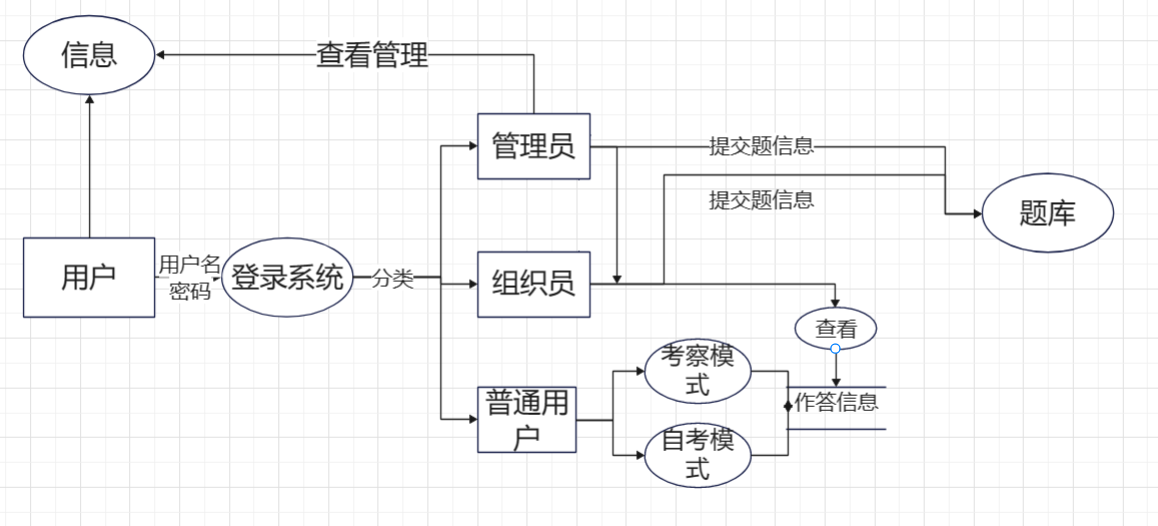


图2.系统数据流图

# 逻辑结构设计

## 关系模式

根据E-R图向关系模型的转换原则，党的基本知识测试系统的E-R图可以转换为下列关系模式：

**1、users(username,password,email,role);**

2、**question\_bank(no,content,option1,option2,option3,option4,answer);**

3、**[learning\_record](http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=sgm&table=learning_record" \o "浏览)(username,score,start\_time,end\_time);**

4、**paper(no,num,start,end);**

**5、inspect(no,username,score,time);**

**6、statistics(login,learn,time);**

## 数据表的详细结构信息

表9.users(用户)数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| username | 唯一标识每位用户 | Varchar | 20 |
| password | 用户登录密码 | Varchar | 20 |
| email | 用户个人邮箱 | Varchar | 20 |
| role | 身份角色 | Varchar | 20 |

表10.question\_bank(题库)数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| No | 唯一标识题号 | Varchar | 20 |
| Content | 题目内容 | Varchar | 200 |
| Option1 | 选项A | Varchar | 20 |
| Option2 | 选项B | Varchar | 20 |
| Option3 | 选项C | Varchar | 20 |
| Option4 | 选项D | Varchar | 20 |
| Answer | 题目答案 | Varchar | 20 |

表11.[learning\_record](http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=sgm&table=learning_record" \o "浏览)(自学自考记录)数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| Username | 唯一标识用户 | Varchar | 20 |
| Score | 自考成绩 | Varchar | 20 |
| Start\_time | 开始时间 | Varchar | 20 |
| End\_time | 结束时间 | Varchar | 20 |

表12.paper(试卷)数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| No | 试卷编号 | Varchar | 20 |
| Num | 试卷中有哪些题目号 | Varchar | 20 |
| Start | 试卷做题开始时间 | Varchar | 20 |
| End | 结束时间 | Varchar | 20 |

表13.inspect(考察记录)数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| No | 试卷编号 | Varchar | 20 |
| Username | 用户名 | Varchar | 20 |
| Score | 成绩 | Varchar | 20 |
| Time | 做题时间 | Varchar | 20 |

表14.statistics(登录/学习次数记录)数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 含义说明 | 类型 | 长度 |
| login | 登录次数 | Int | 11 |
| learn | 学习次数 | Int | 11 |
| time | 时间 | Varchar | 20 |

## 系统结构图（模块结构）

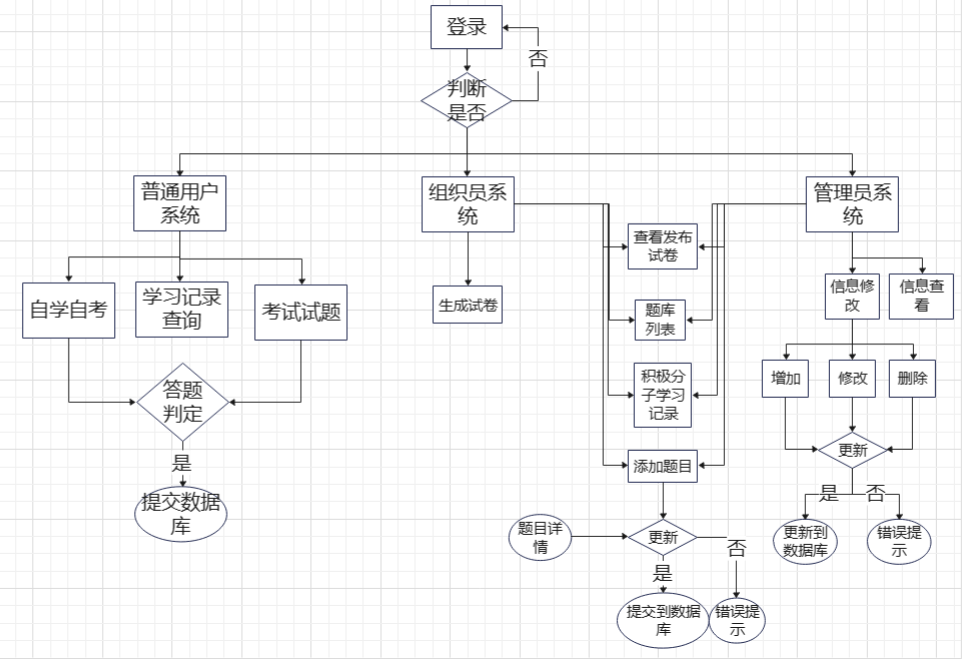


图3.系统结构图

# 物理设计（参照教材）

## 存储安排

确定数据库物理结构的内容要确定数据的存放位置和存储结构（关系、索引、聚簇、日志、备份）和确定系统配置。影响数据存放位置和存储结构的因素既包括硬件环境也包括应用需求（存取时间、存储空间利用率、维护代价）。由于此系统规模较小，所以可以将此系统放于同一磁盘上进行使用。

## 1、存取方法选择

DBMS常用存取方法有顺序存取、散列存取、聚簇存取、索引方法，目前主要是B+树索引方法、聚簇（Cluster）方法和HASH方法。本系统对数据库物理设计的要求不高，采用SQL SERVER 默认分配的物理空间即可满足实践要求，本系统对物理设计并未采取特别设计。

## 2、存取路径建立

数据库数据备份、日志文件备份等由于只在故障恢复时才使用，而且数据量很大，可以考虑与数据库对象（表、索引等）放在不同的磁盘以改进系统的性能。

## 模块设计（IPO表）

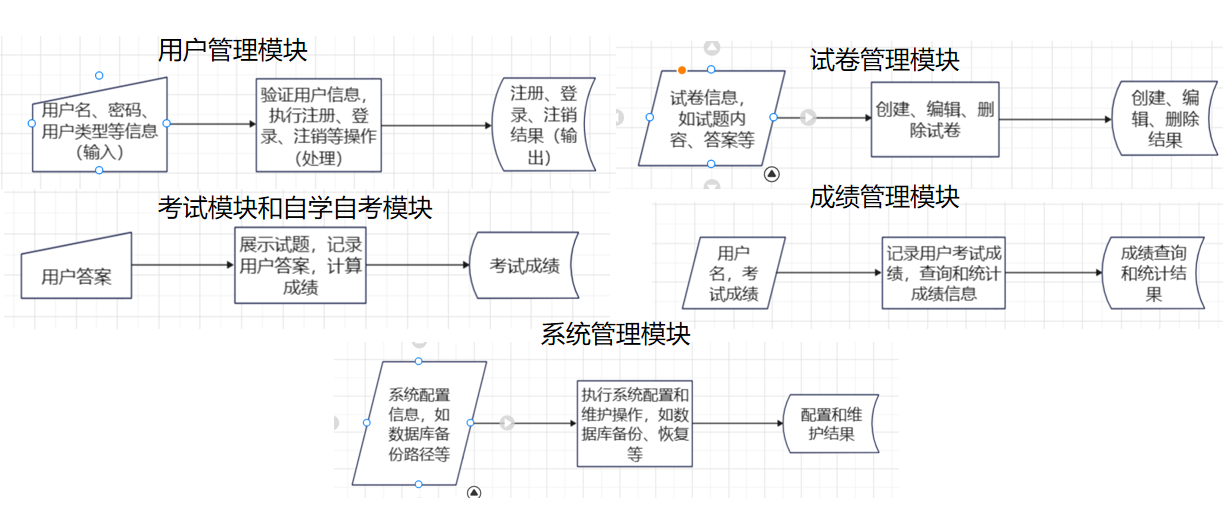


图3.模块ipo设计图

# 数据库实施

## 创建关系模式（列出全部数据表的创建语句）

1. **人员表**

CREATE TABLE user (  
 username VARCHAR(20) PRIMARY KEY,  
 password VARCHAR(20),  
 email VARCHAR(20),  
 role VARCHAR(20)  
);  
**（2）题库表**

CREATE TABLE question\_bank (  
 no VARCHAR(20) PRIMARY KEY,  
 content VARCHAR(200),  
 option1 VARCHAR(20),  
 option2 VARCHAR(20),  
 option3 VARCHAR(20),  
 option4 VARCHAR(20),  
 answer VARCHAR(20)  
);  
**（3）学习记录表**   
CREATE TABLE learning\_record (  
 username VARCHAR(20) PRIMARY KEY,  
 score VARCHAR(20),  
 start\_time VARCHAR(20),  
 end\_time VARCHAR(20)  
);  
**（4）试卷试题信息表**   
CREATE TABLE paper (  
 no VARCHAR(20),  
 num VARCHAR(20),  
 start VARCHAR(20),  
 end VARCHAR(20)  
);  
**（5）试卷考察记录表**   
CREATE TABLE inspect (  
 no VARCHAR(20) PRIMARY KEY,  
 username VARCHAR(20),  
 score VARCHAR(20),  
 time VARCHAR(20)  
);

**（6）登录/学习次数记录表**

CREATE TABLE `statistics` (

`login` int(11) NOT NULL,

`learn` int(11) NOT NULL,

`time` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

## 程序代码

1. **用户登录代码** (利用表单post获取当前登录用户名并进行session变量定义，判断用户身份，并根据用户身份登陆后跳转到不同的界面)

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
// 从表单中获取用户名  
$username = $\_POST["username"];   
$role=$\_POST["role"];  
// 读取session中的用户名信息  
session\_start();  
$\_SESSION['username'] = $username;  
$\_SESSION['role'] = $role;  
$password = $\_POST["password"];  
// 检查用户是否存在  
$sql = "SELECT \* FROM users WHERE username='$username' AND password='$password'";  
$result = $conn->query($sql);  
if ($result->num\_rows ==1) {  
 $\_SESSION["loggedin"] = true;   
 while($ro = mysqli\_fetch\_assoc($result)) {  
$confirm\_role=$ro["role"];}  
// 页面跳转代码  
 if($role=="admin"&& $role==$confirm\_role)  
 {header("Location: manage\manage.php");  
 exit();}  
 elseif($role=="organizer"&& $role==$confirm\_role)  
 {header("Location: organizer\organizer.php");  
 exit();}  
 elseif($role=="user"&& $role==$confirm\_role)  
 {header("Location: users\user.php");  
exit();}

1. **用户自学自考功能实现**（使用随机数组生成自考的题号数组，使用html表单标签在网页中展示，并由用户选择玩答案后提交表单，进行成绩判定和答题详情展示以及答题详情数据插入数据库，此处展示由随机数组生成到自考题展示代码）

<form action="submit.php" method="POST" style="">

<h1>自学自考试题</h1>

<?php

// retrieve questions from database

$query = "SELECT \* FROM question\_bank order by no";

$result = mysqli\_query($conn, $query);

$sqll = "SELECT count(no) FROM question\_bank";

$res = mysqli\_query($conn, $sqll);

while ($row = mysqli\_fetch\_assoc($res)) {

$num=$row["count(no)"];

}

//生成随机数组

$random\_numbers = array();

while (count($random\_numbers) < 10) {

$random\_number = rand(1, $num);

if (!in\_array($random\_number, $random\_numbers)) {

$random\_numbers[] = $random\_number;

}

}

$i = 1;

foreach ($random\_numbers as $number) {

if($i<=10){

// display question number and content

while ($row = mysqli\_fetch\_assoc($result)) {

if($number==$row["no"]){

echo "<p>" . $i . ". " . $row["content"] . "</p>";

// display answer choices as radio buttons

echo "<div>";

echo "<input type='radio' name='answer-" . $row["no"] . "' value='A'>" . $row["option1"] . "<br>";

echo "<input type='radio' name='answer-" . $row["no"] . "' value='B'>" . $row["option2"] . "<br>";

echo "<input type='radio' name='answer-" . $row["no"] . "' value='C'>" . $row["option3"] . "<br>";

if ($row["option4"]!=NULL) {echo "<input type='radio' name='answer-" . $row["no"] . "' value='D'>" . $row["option4"] . "<br>";}

echo "</div>";

}

}

$i++;

$query = "SELECT \* FROM question\_bank order by no";

$result = mysqli\_query($conn, $query);

}

}

?>

<input type="submit" name="submit" class="sub" value="提交">

</form>

1. **组织员发布试卷功能实现**（用form表单完成该功能，从题库中把所有的题目依次排列展示出来，并提供按钮可进行选择，组织员选择完题目后选择时间，最后使用form的button进行提交）

<form action="submit.php" method="POST" onsubmit="return validateForm();">

<h1>试卷试题</h1>

<?php

// retrieve questions from database

$query = "SELECT \* FROM question\_bank order by no";

$result = mysqli\_query($conn, $query);

$i=1;

while ($row = mysqli\_fetch\_assoc($result)) {

?>

<table class="ti">

<td>

<?php

echo "<input type='radio' name='".$i."' value='".$row["no"]."'>" . $row["content"] . "</p>";

// display answer choices as radio buttons

$i++;

?>

</td>

</table>

<?php

}

?>

</div>

<div class="form-wrapper">

<label for="start-time">试卷起始时间：</label>

<input type="datetime-local" id="start-time" name="start-time" required><br>

<label for="end-time">试卷结束时间：</label>

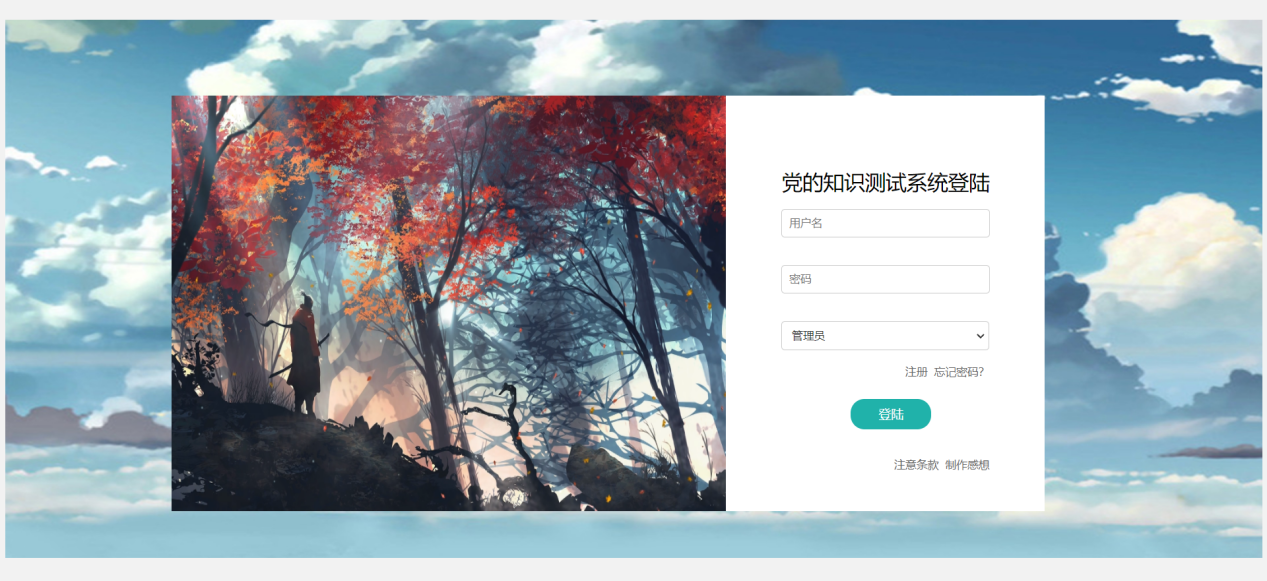
<input type="datetime-local" id="end-time" name="end-time" required><br>

<input type="submit" class="sub" value="发布试卷">

</div>

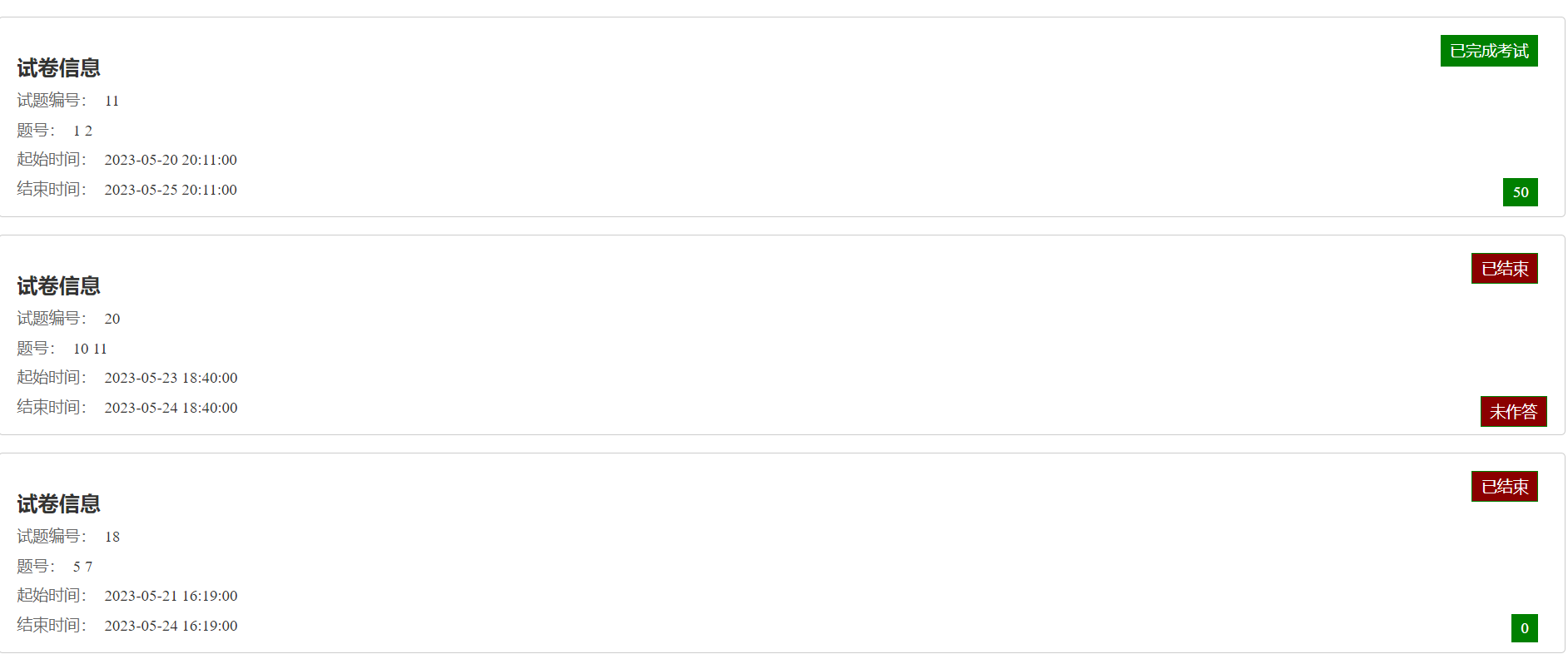
</form>

## 测试

1. **初始界面测试图**
2. 用户界面（以管理员为例）

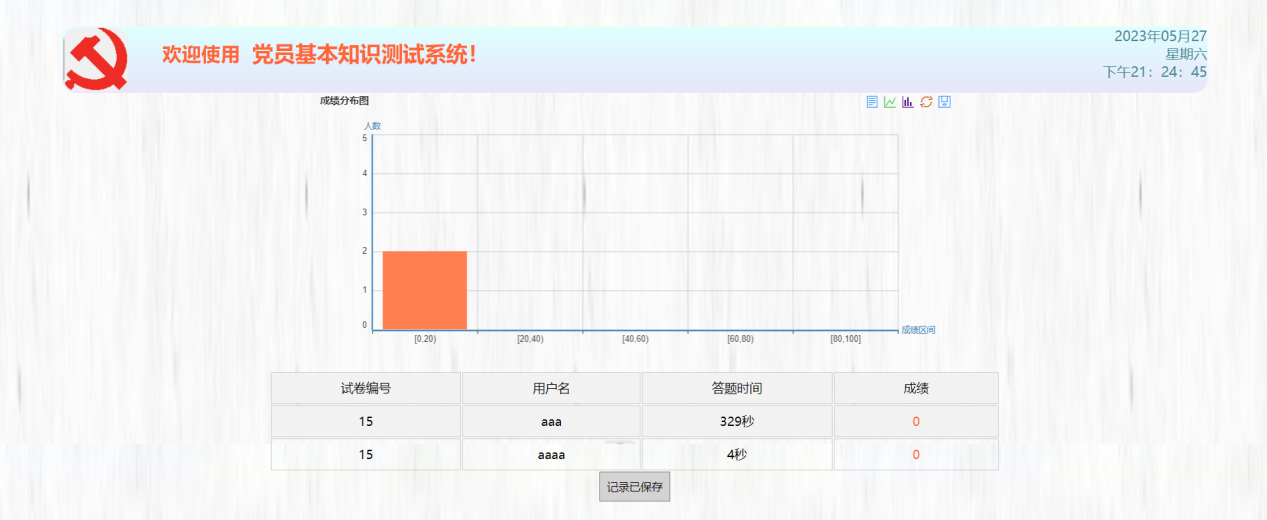


1. 试卷界面



1. 自学自考答题详情界面

（5）考察统计详情界面



# 系统设计相关代码（列举代表水平的部分）

## 成绩判定及答题详情展示的代码

//成绩判定

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get the answers from the form

$results = array();

foreach ($\_POST as $name => $value) {

$results[$name] = $value;

}

if(count($results)!=11){

echo "<script>window.alert('题目未达完！');</script>";

exit();

}

// Get the correct answers from the database

$sql = "SELECT \* FROM question\_bank";

$result = mysqli\_query($conn, $sql);

// Check if there are any results

if (mysqli\_num\_rows($result) > 0) {

// Initialize the score to 0

$score = 0;

// Loop through each question

foreach ($results as $key => $value) {

while ($row = mysqli\_fetch\_assoc($result)) {

// Check if the user's answer matches the correct answer

$num = 'answer-' . $row["no"];

if($key==$num){

$user\_answer=$value;

$correct\_answer = $row["answer"];

//在此进行题目详解

?>

<div style="text-align: center;margin-left: 100px;">

<table style="text-align: center">

<tbody>

<tr>

<td><?php echo $row["content"]; ?></td>

<td><?php echo '所选:'.$user\_answer; ?></td>

<td><?php echo '答案：'.$correct\_answer; ?></td>

<td><?php if($user\_answer==$correct\_answer){ ?>

<p style="color:green">&#10003;</p><?php

}else{ ?> <p style="color:#ff6633">&#10007;</p> <?php } ?></td>

</tr>

</tbody>

</table>

</div>

<?php

if ($user\_answer == $correct\_answer) {

// Increment the score if the answer is correct

$score += 10;

}

// Move on to the next question

}

}

## 试卷设置时间的选择的代码（通过javascript脚本实现时间的选择而不用组织员进行手动输入）

<script>

function getCurrentDateTime() {

var now = new Date();

var month = (now.getMonth() + 1).toString().padStart(2, '0');

var day = now.getDate().toString().padStart(2, '0');

var currentDateTime = now.getFullYear() + '-' + month + '-' + day;

return currentDateTime;

}

</script>

<script>

// 设置起始时间的初始值为当前日期

document.getElementById("start-time").value = getCurrentDateTime();

// 设置结束时间的初始值为当前日期的后一天

var tomorrow = new Date();

tomorrow.setDate(tomorrow.getDate() + 1);

document.getElementById("end-time").value = tomorrow.toISOString().slice(0, -8);

</script>

<div class="form-wrapper">

<label for="start-time">试卷起始时间：</label>

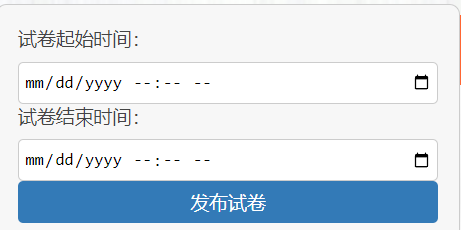
<input type="datetime-local" id="start-time" name="start-time" required><br>

<label for="end-time">试卷结束时间：</label>

<input type="datetime-local" id="end-time" name="end-time" required><br>

<input type="submit" class="sub" value="发布试卷">

</div>



## 统计表格的绘制（使用了 ECharts 库来创建一个柱状图，定义了一个DOM 元素，然后引入了 ECharts 库。）

<div style="margin:0; margin-bottom: 0 !important;">

<!-- 为ECharts准备一个具备大小（宽高）的Dom -->

<div id="main" style="width:50%;height:400px;margin: 0 auto"></div>

<!-- ;margin-top:80px;控制距离顶部距离 -->

<!-- ECharts单文件引入 -->

<script src="http://echarts.baidu.com/build/dist/echarts.js"></script>

<script type="text/javascript">

// 路径配置

require.config({

paths: {

echarts: 'http://echarts.baidu.com/build/dist'

}

});

// 使用

require(

[ 'echarts', 'echarts/chart/bar', 'echarts/chart/line' ],

function (ec) {

// 基于准备好的dom，初始化echarts图表

var myChart = ec.init(document.getElementById('main'));

var option = {

title: {text: '成绩分布图',textStyle: { //主标题文本样式{"fontSize": 18,"fontWeight": "bolder","color": "#333"}

fontSize: 14,fontStyle: 'normal',fontWeight: 'bold',}, },

tooltip: {trigger: 'axis'},

toolbox: {

show: true,feature: {dataView: {show: true,readOnly: false},

magicType: {

show: true,type: ['line', 'bar'] },

restore: {show: true},

saveAsImage: {

show: true}}},

calculable: true,

xAxis: [{

type: 'category',data: ['[0,20)', '[20,40)', '[40,60)', '[60,80)','[80,100]'],name: '成绩区间',position: 'left' }],

yAxis: [{

type: 'value',name: '人数',position: 'left' }],

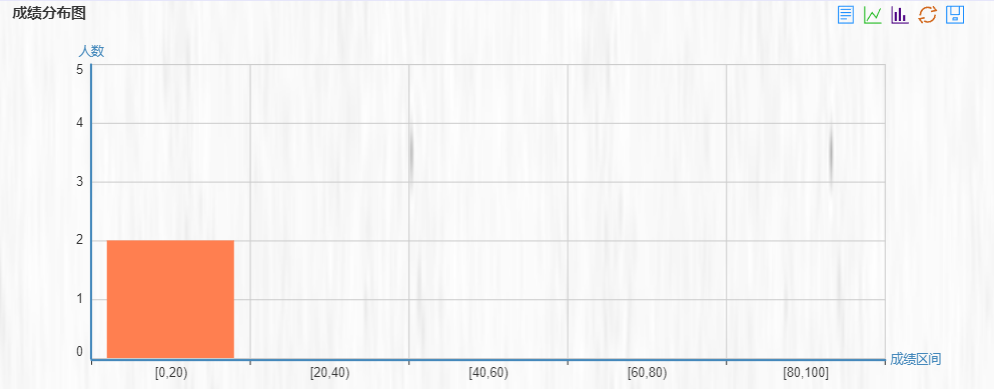
series: [{

name: '人数',type: 'bar',data: [<?php echo $c1?>,<?php echo $c2?>,<?php echo $c3?>,<?php echo $c4?>,<?php echo $c5?>],color: '#CC0066' }, ]};

// 为echarts对象加载数据

myChart.setOption(option); });

</script>



# 感想及心得体会

这是一次让我受益极大的一门课程，在上半学期学习了web应用开发了解了php连接数据库以及html页面展示后开设的这门课程让我有了实践的地方，因为有了基础，网站的设计非常迅速，在热情的驱使下，不到2个周就几乎完成了所有功能的实现及初始美化，为了网站更加美观，我又花了半周参考了其他网站的设计终于完成，我对我的课设非常自信以及自豪，在网站的制作感想中我如下写到：

<p > 数据是一粒粒的种子~~</p>

<p> php数据处理方法是雨水~~</p>

<p>  没有数据支撑的前端技术华而不实，没有前端制成的网页有的只是冰冷的数据</p>

<p>  当土壤与种子相结合，再有了雨水的浇灌，它们的杰作才是那么伟大，茂盛的丛林、欣欣向荣的自然区，无疑带来美的感受---</p>

这是我对于这门课学到的真实写照，它让我明白了一个设计前端以及后端的重要性，一个好的网站应该是前端布局优美，后端数据类型多样丰富，才能实现一个有内容、有享受的好网站。