# 学生选课管理系统

### 概述

本次作业实现的是一个简易的学生选课管理系统，基于如下开发环境：

* Web服务器：Apache
* 开发语言：HTML、JavaScript、PHP
* 数据库管理系统：MySQL

开发流程如下：

用户通过浏览器端进行各种操作，JS将数据打包传送给后端的PHP，PHP根据不同的请求进行相应的响应，从数据库中取得数据并返回给前端。

### 数据库表的定义

学生选课管理系统共包含四个表，分别是教师表、学生表、课程表、学生选课信息表，各个表的定义如下：

1. **教师表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| TeacherId | Int(10) | No | PRI | NULL |  |
| T TeacherName | Varchar(10) | No |  | NULL |  |
| TeacherGender | Varchar(10) | No |  | NULL |  |

1. **学生表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| StudentName | Varchar(10) | StudentId | Int(10) | NULL |  |
| StudentGender | Varchar(10) | YES |  | NULL |  |
| Password | Varchar(50) | YES |  | NULL |  |

1. **课程表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| TeacherId | Int(10) | NO | PRI | NULL |  |
| CourseId | Varchar(20) | NO | PRI | NULL |  |
| CourseName | Float | NO |  | NULL |  |
| CoourseCredit | Smllint(6) | NO |  | NULL |  |
| CourseClass | Smallint(6) | NO |  | NULL |  |
| Capacity | Int(10) | NO |  | NULL |  |
| Remain | Smallint(6) | YES |  | NULL |  |

1. **学生选课表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| StudentId | Int(10) | NO | PRI | NULL |  |
| CourseId | Int(10) | NO | PRI | NULL |  |
| TeacherId | Int(10) | NO | PRI | NULL |  |
| Grade | Smallint(6) | YES |  | NULL |  |

其中，功能较为完善的有学生表和选课表，次之是学生选课表，最后是教师表。

### PHP与MySQL交互

采用PHP自带MySQL接口，实现增删改查等功能。

1. **数据库的连接**
2. $conn = **new** mysqli(“localhost”, “username”, “password”, “database”)
3. **增删改查**

下面以选课表为例演示对选课表的增删改查

* 增添新课程

1. <?php
2. $conn = **new** mysqli("localhost", "manager", "\*\*\*\*\*\*\*", "SEMS");  //连接数据库SEMS
3. $sql = "INSERT IGNORE INTO
4. tb\_course(CourseId, CourseName, CourseCredit, CourseClass, Capacity, TeacherId, Remain)
5. VALUES(".$courseId.", '".$courseName."', ".$courseCredit.", ".$courseClass.", ".$capacity.", ".$teacherId.", ".$capacity.");";
6. $conn->query($sql);  //执行mysql语句
7. ?>

程序说明：首先，HTML通过表单提交增添课程所需要的教师号、课程号、课程名等必要数据，php通过POST方法获取这些数据，然后通过query接口执行插入语句，从而添加课程成功。

* 删除课程

1. <?php
2. $conn = **new** mysqli("localhost", "manager", "\*\*\*\*\*\*\*", "SEMS");  //连接数据库SEMS
3. $sql = "DELETE FROM tb\_course WHERE CourseId=".$courseId." and TeacherId=".$teacherId."";
4. $conn->query($sql);
5. ?>

程序说明：HTML通过表单提交删除课程所需要的课程号和教师号，php通过GET方法获取数据，然后通过query接口执行删除语句，从而从数据库中删除课程。

* 修改课程

1. <?php
2. $conn = **new** mysqli("localhost", "manager", "\*\*\*\*\*\*\*", "SEMS");  //连接数据库SEMS
3. $sql = "UPDATE tb\_course SET Capatity=".$update." WHERE CourseId=".$courseId."";
4. $conn->query($sql);
5. ?>

程序说明：支持修改课程容量，HTML通过表单提交课程号和课程容量，php通过GET方法获取数据，接着使用query接口执行课程容量更新操作，并确保更新的课程容量不得小于已选课程人数。

* 查询课程

1. <?php
2. $conn = **new** mysqli("localhost", "manager", "\*\*\*\*\*\*\*", "SEMS");  //连接数据库SEMS
3. $sql = "SELECT \* FROM tb\_course WHERE CourseName='".$courseName."'";
4. $conn->query($sql);
5. ?>

程序说明：HTML通过表单提交课程名，php通过GET方法获取数据，接着使用query接口执行课程查询操作，并返回结果。

* 添加个人信息

1. $ret = mysqli\_query($sql,
2. "insert into tb\_student (studentId,studentName,studentGender,password)".
3. " values($id, '$name', $gender, '$password')");

程序说明：HTML通过表单提交个人信息，接着使用insert接口执行课程查询操作，并返回结果。

* 查询个人信息

1. $ret = mysqli\_query($sql,
2. "update tb\_student set studentName='$name', studentGender=$gender ".
3. "where studentId=$id");

程序说明：HTML通过表单提交个人信息，接着使用update接口执行课程查询操作，并返回结果。

* 查询全体信息

1. $result = mysqli\_query($sql,
2. "select studentId, studentName, studentGender from tb\_student");

程序说明：HTML通过表单提交个人信息，接着使用update接口执行课程查询操作，并返回结果。

* 删除个人信息

1. $ret = mysqli\_query($sql,
2. "delete from tb\_student where studentId=".$id);

程序说明：HTML通过表单提交个人信息，接着使用update接口执行课程查询操作，并返回结果。

### 功能展示

见视频 展示.mp4