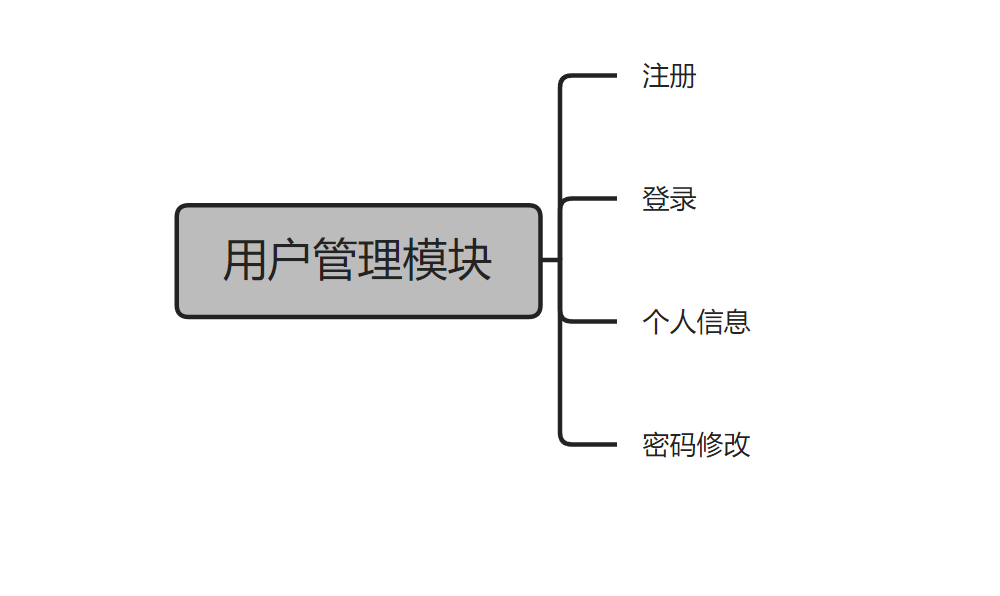
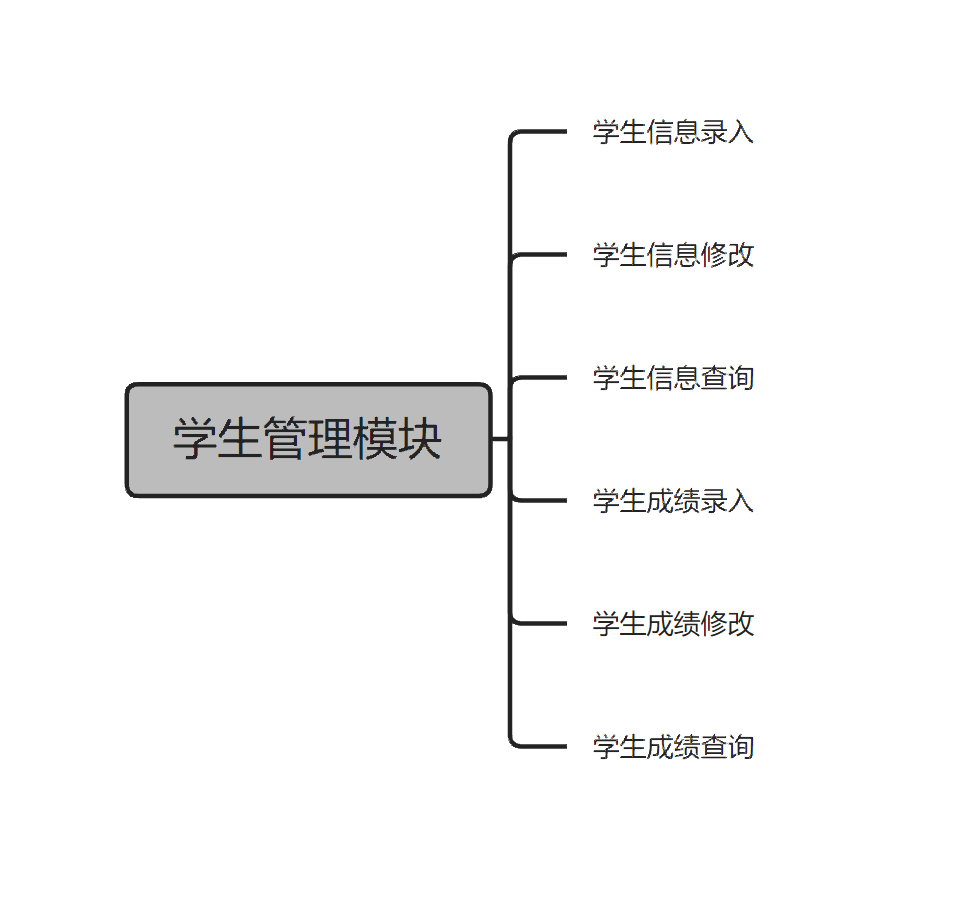
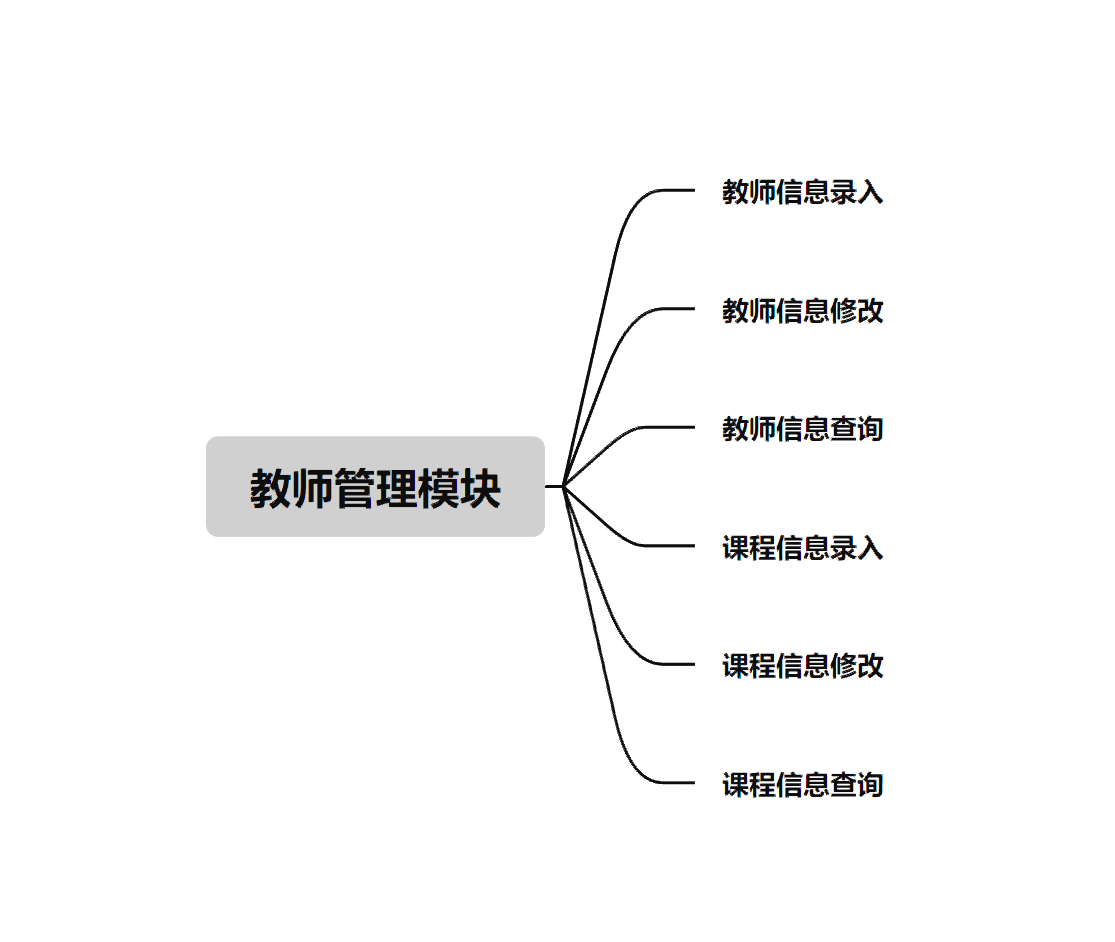
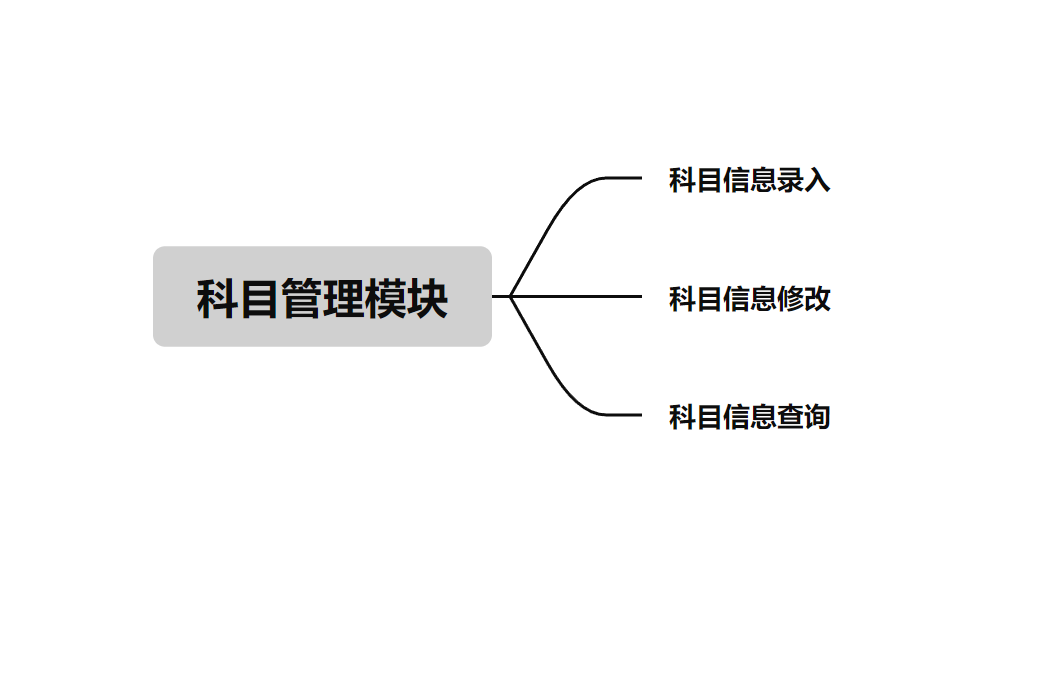
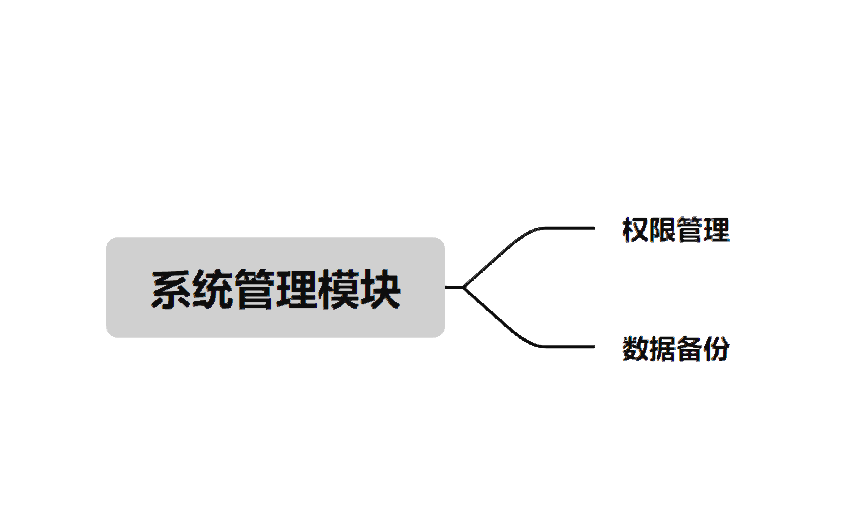
### 1.系统功能模块及框图











#### 1.1.1 用户管理模块

* 注册
* 登录
* 个人信息管理
* 密码修改

#### 1.1.2 学生管理模块

* 学生信息录入
* 学生信息修改
* 学生信息查询
* 学生成绩录入
* 学生成绩修改
* 学生成绩查询

#### 1.1.3 教师管理模块

* 教师信息录入
* 教师信息修改
* 教师信息查询
* 课程信息录入
* 课程信息修改
* 课程信息查询

#### 1.1.4 科目管理模块

* 科目信息录入
* 科目信息修改
* 科目信息查询

#### 1.1.5 系统管理模块

* 权限管理
* 数据备份与恢复
* 系统日志查看

### 2. 数据库设计说明

#### 2.1 用户表（user）

| **字段名** | **数据类型** | **描述** |
| --- | --- | --- |
| id | INT | 用户ID |
| username | VARCHAR(50) | 用户名 |
| password | VARCHAR(255) | 密码（加密存储） |
| status | INT | 用户状态 |

#### 2.2 学生表（student）

| **字段名** | **数据类型** | **描述** |
| --- | --- | --- |
| id | INT | 学生ID |
| studentId | VARCHAR(20) | 学号 |
| name | VARCHAR(50) | 姓名 |
| className | VARCHAR(50) | 班级 |
| birthday | DATE | 生日 |
| sex | VARCHAR(10) | 性别 |
| nation | VARCHAR(20) | 民族 |

#### 2.3 教师表（teacher）

| **字段名** | **数据类型** | **描述** |
| --- | --- | --- |
| id | INT | 教师ID |
| teacherId | VARCHAR(20) | 工号 |
| name | VARCHAR(50) | 姓名 |

#### 2.4 课程表（course）

| **字段名** | **数据类型** | **描述** |
| --- | --- | --- |
| id | INT | 课程ID |
| courseId | VARCHAR(20) | 课程代码 |
| teacherId | INT | 授课教师ID |
| subjectId | INT | 开设科目ID |

#### 2.5 科目表（subject）

| **字段名** | **数据类型** | **描述** |
| --- | --- | --- |
| id | INT | 科目ID |
| subjectId | VARCHAR(20) | 科目代码 |
| teacherId | INT | 负责教师ID |
| subjectname | VARCHAR(50) | 科目名称 |
| content | TEXT | 科目简介 |

#### 2.6 成绩表（score）

| **字段名** | **数据类型** | **描述** |
| --- | --- | --- |
| id | INT | 成绩ID |
| studentId | INT | 学生ID |
| term | VARCHAR(20) | 学期 |
| subjectId | INT | 科目ID |
| mark | DECIMAL(5,2) | 分数 |

### **3. 用户管理模块核心程序**

#### 3.1 注册功能 (registerdo.php)

在注册功能中，我们首先接收用户通过注册页面提交的用户名和密码。使用**sha1**函数对密码进行哈希处理，增加安全性。然后，通过SQL语句将用户信息插入到**user**表中，实现用户的注册功能。在注册成功时，通过JavaScript弹窗提示用户注册成功，并跳转到登录页面；注册失败时，同样通过弹窗提示用户注册失败，并返回到注册页面。

#### 3.2 登录功能 (logindo.php)

登录功能中，我们接收用户输入的用户名和密码，并通过SQL查询判断用户是否存在于数据库中。如果查询结果非空，表示用户存在且密码正确，将用户名存入**$\_SESSION**中，表示用户已登录。然后跳转到系统首页。若查询结果为空，则提示用户用户名或密码错误，回到登录页面。

### **4. 学生管理模块核心程序**

#### 4.1 学生信息录入 (insertdo.php)

学生信息录入功能接收从前端提交的学生信息，包括学号、姓名、班级、生日、性别、民族等。通过SQL语句将这些信息插入到**student**表中，完成学生信息的录入。成功时弹窗提示录入成功并跳转到系统首页，失败时提示录入失败并返回录入页面。

#### 4.2 学生信息修改 (editdo.php)

学生信息修改功能接收前端传来的修改后的学生信息，通过SQL语句更新**student**表中相应的记录。成功时弹窗提示修改成功并跳转到系统首页，失败时提示修改失败并返回修改页面。

### **5. 教师管理模块核心程序**

#### 5.1 教师信息录入 (insertteacherdo.php)

教师信息录入功能接收前端传递的教师信息，包括教师编号和姓名。通过SQL语句将这些信息插入到teacher表中，实现教师信息的录入。成功时弹窗提示录入成功并跳转到教师管理页面，失败时提示录入失败并返回录入页面。

在每个核心程序中，我们使用了数据库连接、SQL语句执行、成功与失败的提示等基本的PHP编程技巧。这些程序实现了系统中关键功能，确保了用户能够顺利进行注册、登录、学生信息录入和修改、教师信息录入等操作。

窗体顶端

窗体底端

## 6. 总结及体会

在开发课程系统的过程中，我们充分考虑了用户管理、学生管理、教师管理、科目管理和系统管理等多个模块，确保系统的完整性和稳定性。数据库设计采用了关系型数据库，通过合理的表结构和关联建立了各模块之间的关联关系，保证了数据的一致性和完整性。

在核心程序的设计上，我们采用了PHP语言，结合MySQL数据库，通过PDO连接数据库，防范了SQL注入等安全问题。同时，我们使用了一些前端插件，提升了用户交互体验。

通过本次开发，我们深刻体会到团队协作的重要性，每个成员在不同模块的开发中发挥了各自的优势，共同推动了整个项目的进展。

## 7. 分工说明

* **成员A：** 负责用户管理模块、学生管理模块；
* **成员B：** 负责教师管理模块、科目管理模块；
* **成员C：** 负责系统管理模块、核心程序的优化及前端页面设计。

以上分工仅为简化说明，实际工作中可能有交叉和协同工作。

## 8. 结语

通过这次综合课程系统开发，我们不仅提升了对数据库设计、PHP编程和团队协作的理解，同时也深入学习了如何设计和实现一个完整的Web应用系统。这次经验将对我们今后的学习和工作产生深远的影响。