学号：102301116 姓名：俞烨 专业：计算机科学与技术 班级：计算机类01班 《数据库应用实践》实验三：数据库应用系统的开发

实验环境：openEuler+openGauss数据库+windows/linux部署

实验内容：

一、系统选题与工作情况说明

选题：学生选课与成绩管理系统（B/S 架构）。

角色与功能：

学生：查看可选课程、选课/退选、查看所选课程成绩、修改密码。

教师：查看自己授课的课程、查看课程学生名单、录入/修改成绩、修改密码。

管理员：学生/教师/课程/选课全量 CRUD，Excel 批量导入课程与学生名单，导出课程成绩 Excel，用户与权限管理。

技术栈：

前端：Vue 3 + Vite，Fetch API 调用后端 REST。

后端：Flask + psycopg2 + openGauss；可选 SSH 隧道（sshtunnel/paramiko）。

数据库：openGauss（兼容 PostgreSQL 语法）。

主要工作概览：

设计与实现用户认证与基于角色的权限控制（Session）。

数据库概念与逻辑模型设计、DDL 与索引、约束实现。

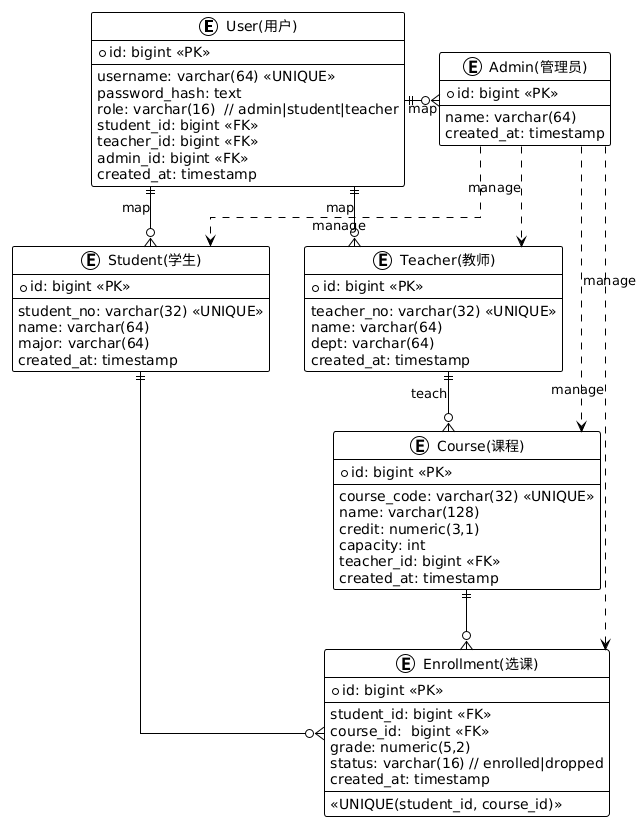
完整 REST API（学生/教师/管理员）。

Excel 导入课程/学生/选课、导出成绩（中文表头，文件名“课程-老师-时间.xlsx”）。

前端三大界面实现与美化（紧凑简约设计）。

二、概念数据模型（E-R 图）

实体：用户(User)、管理员(Admin)、学生(Student)、教师(Teacher)、课程(Course)、选课(Enrollment)。



三、系统中每张表的说明

**users**

* username：登录名，管理员为 admin；学生/教师默认规则支持“学号/工号”。
* password\_hash：口令哈希（后端统一校验）。
* role：student/teacher/admin。
* student\_id/teacher\_id：与真实身份关联（其一为空）。

**students**

* student\_no：学号，唯一。
* name/major：基本信息。

**teachers**

* teacher\_no：工号，唯一。
* name/dept：基本信息。

**courses**

* course\_code：课程号，唯一。
* name/credit/capacity：课程属性。
* teacher\_id：授课教师（FK）。

**enrollments**

* student\_id/course\_id：复合唯一约束防重复选课。
* grade：成绩，可空。
* status：enrolled/dropped，用于退选保留历史。

数据定义语言（DDL）（openGauss/PostgreSQL 语法）：

CREATE TABLE IF NOT EXISTS users (

id bigserial PRIMARY KEY,

username varchar(64) UNIQUE NOT NULL,

password\_hash text NOT NULL,

role varchar(16) NOT NULL CHECK (role IN ('admin','student','teacher')),

student\_id bigint REFERENCES students(id) ON DELETE SET NULL,

teacher\_id bigint REFERENCES teachers(id) ON DELETE SET NULL,

created\_at timestamp DEFAULT now()

);

CREATE TABLE IF NOT EXISTS students (

id bigserial PRIMARY KEY,

student\_no varchar(32) UNIQUE NOT NULL,

name varchar(64) NOT NULL,

major varchar(64),

created\_at timestamp DEFAULT now()

);

CREATE TABLE IF NOT EXISTS teachers (

id bigserial PRIMARY KEY,

teacher\_no varchar(32) UNIQUE NOT NULL,

name varchar(64) NOT NULL,

dept varchar(64),

created\_at timestamp DEFAULT now()

);

CREATE TABLE IF NOT EXISTS courses (

id bigserial PRIMARY KEY,

course\_code varchar(32) UNIQUE NOT NULL,

name varchar(128) NOT NULL,

credit numeric(3,1) DEFAULT 0,

capacity int DEFAULT 50,

teacher\_id bigint REFERENCES teachers(id) ON DELETE SET NULL,

created\_at timestamp DEFAULT now()

);

CREATE TABLE IF NOT EXISTS enrollments (

id bigserial PRIMARY KEY,

student\_id bigint NOT NULL REFERENCES students(id) ON DELETE CASCADE,

course\_id bigint NOT NULL REFERENCES courses(id) ON DELETE CASCADE,

grade numeric(5,2),

status varchar(16) DEFAULT 'enrolled' CHECK (status IN ('enrolled','dropped')),

created\_at timestamp DEFAULT now(),

CONSTRAINT uq\_enroll UNIQUE (student\_id, course\_id)

);

-- 常用索引

CREATE INDEX IF NOT EXISTS idx\_course\_teacher ON courses(teacher\_id);

CREATE INDEX IF NOT EXISTS idx\_enroll\_student ON enrollments(student\_id);

CREATE INDEX IF NOT EXISTS idx\_enroll\_course ON enrollments(course\_id);

四、系统运行环境配置与关键语句

环境

OS：Windows/Linux 开发机 + 云服务器 Linux（openEuler 等）

Python 3.10+，Node 18+，openGauss 5.x

后端依赖：Flask、psycopg2-binary、sshtunnel、paramiko、openpyxl、pandas

安装与运行（Windows\Linux）：

cd /path/to/repository

# 后端

python -m venv .venv

.venv/Scripts/Activate.ps1（source .venv/Scripts/activate）

pip install -r requirements.txt

cd backend

python main.py

# 前端

cd frontend/vue

npm install

npm run dev

数据库连接关键语句（后端简化片段）：

# 连接池 + 可选 SSH 隧道

from psycopg2 import pool

from sshtunnel import SSHTunnelForwarder

if os.getenv('OG\_SSH\_TUNNEL') == 'true':

tunnel = SSHTunnelForwarder(

(os.getenv('OG\_SSH\_HOST'), int(os.getenv('OG\_SSH\_PORT', 22))),

ssh\_username=os.getenv('OG\_SSH\_USER'),

ssh\_password=os.getenv('OG\_SSH\_PASSWORD'),

remote\_bind\_address=(os.getenv('OG\_TUNNEL\_REMOTE\_HOST','127.0.0.1'),

int(os.getenv('OG\_PORT', 5432)))

)

tunnel.start()

host, port = '127.0.0.1', tunnel.local\_bind\_port

else:

host, port = os.getenv('OG\_HOST'), int(os.getenv('OG\_PORT', 5432))

pgpool = pool.SimpleConnectionPool(

1, 10,

host=host, port=port,

dbname=os.getenv('OG\_DBNAME'),

user=os.getenv('OG\_USER'),

password=os.getenv('OG\_PASSWORD'),

)

前后端跨域与会话

Flask：CORS 指定 origin，开启 supports\_credentials；Session SameSite=Lax + HttpOnly。

前端 fetch：统一携带 credentials: 'include'。

Excel 导入/导出（关键说明）

导入工作簿须包含 Sheet“courses”，可选“students”“enrollments”。

导出命名：<课程名称>-<授课老师>-<导出时间>.xlsx，中文表头。

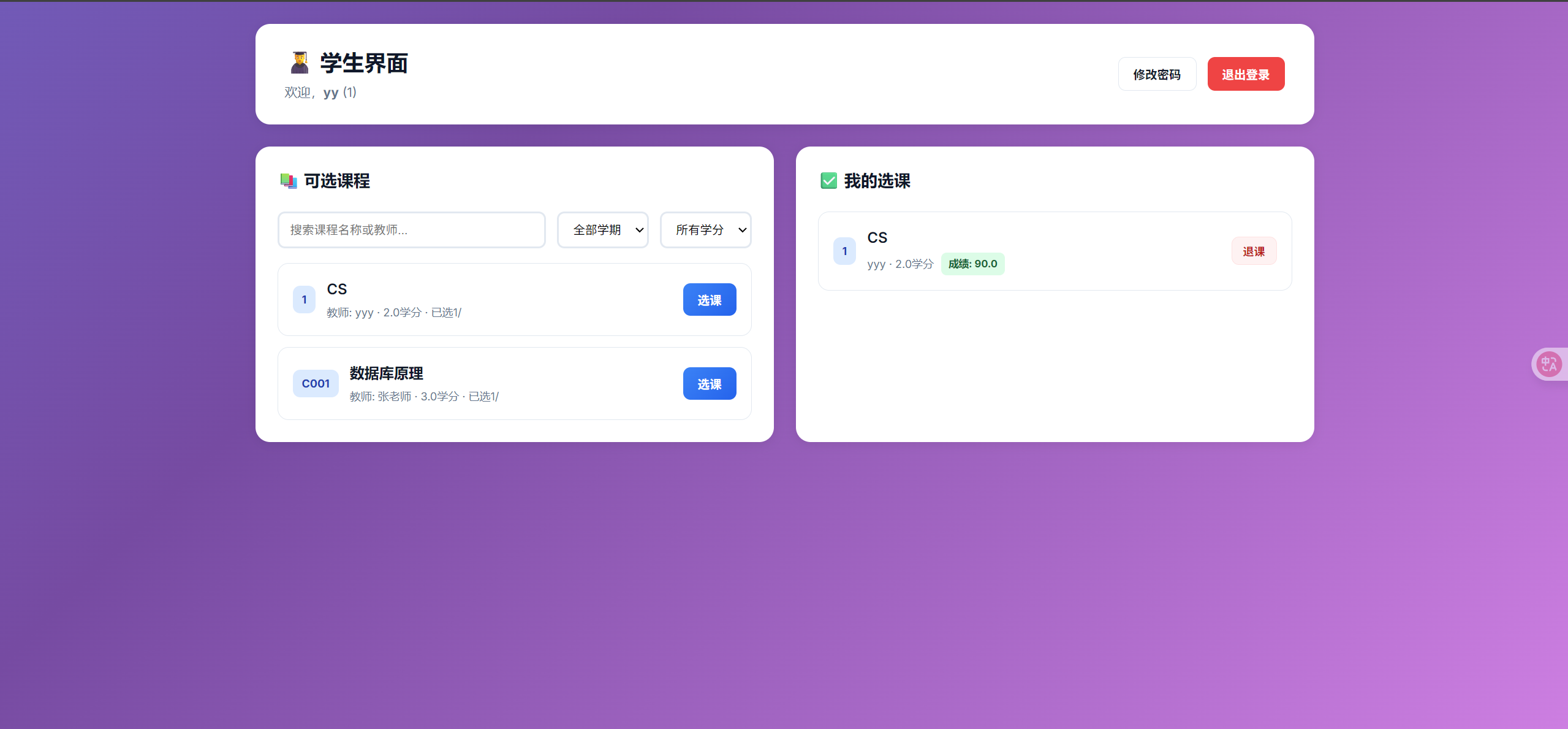
示例生成脚本：backend/scripts/generate\_sample\_excel.py（一键生成 sample\_import.xlsx）。

五、系统主要功能界面

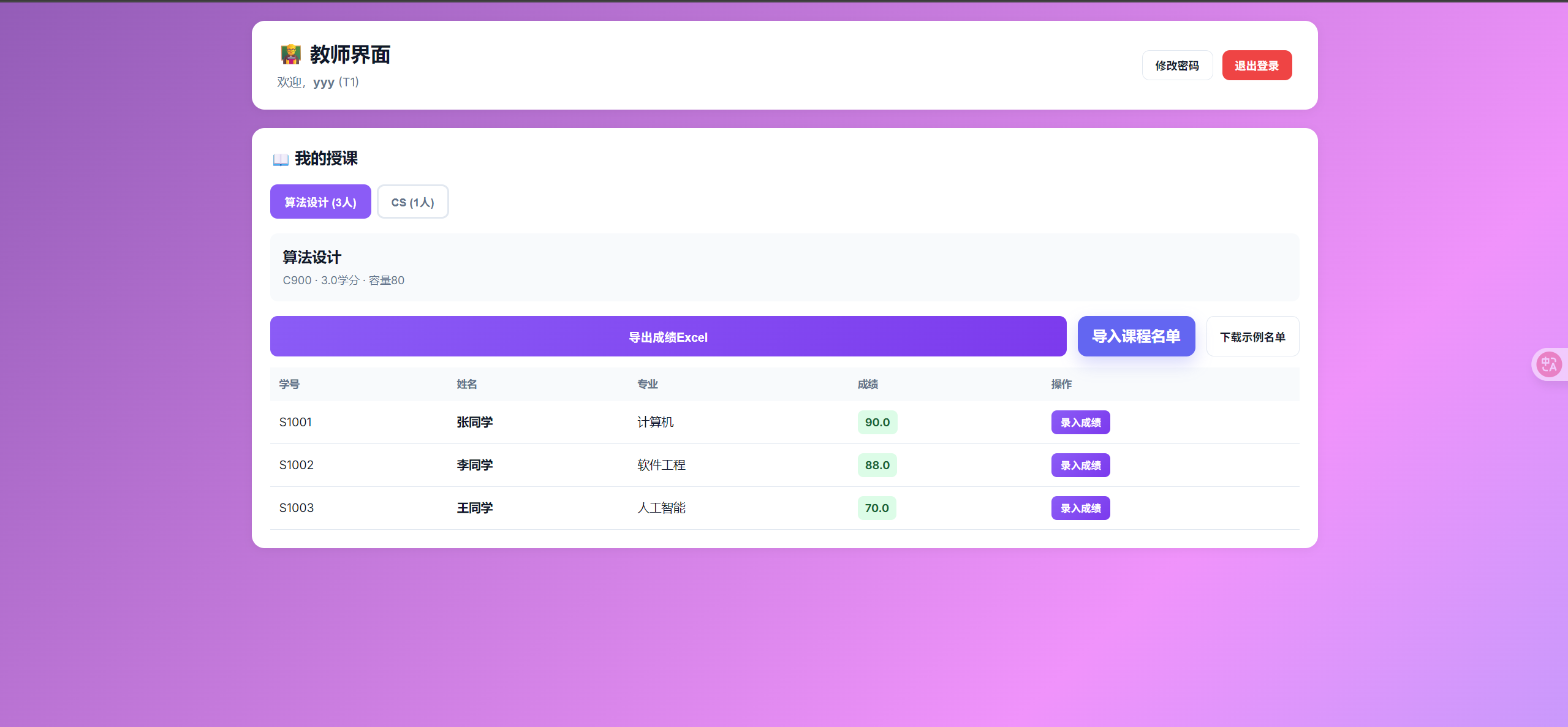
登录页：支持三角色登录与修改密码。



学生页：左侧“可选课程”，右侧“我的选课”；支持选课、退选、显示成绩。



教师页：展示“我的授课”卡片；进入课程可查看学生名单与录入/修改成绩。



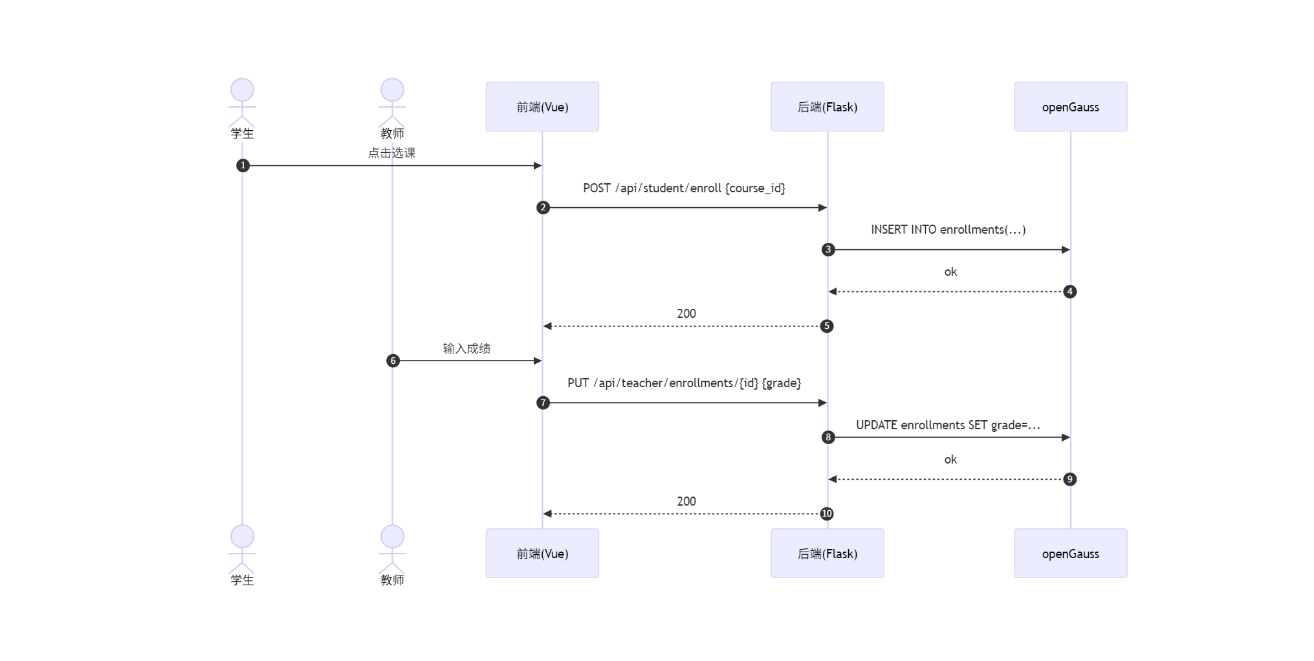
管理员页：学生/教师/课程/选课 CRUD；Excel 导入课程与学生名单；导出课程成绩。







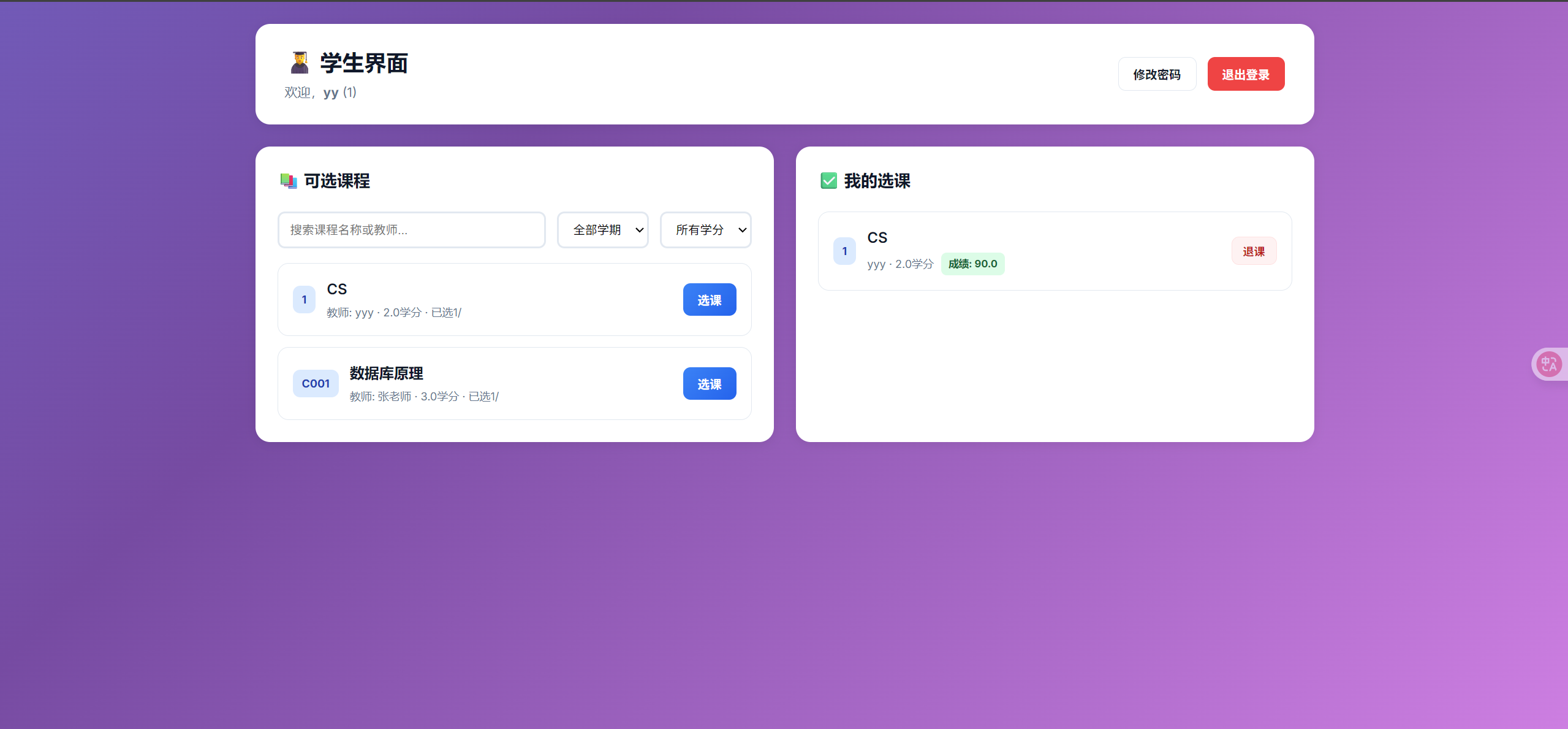
核心交互流程（选课/录分）：



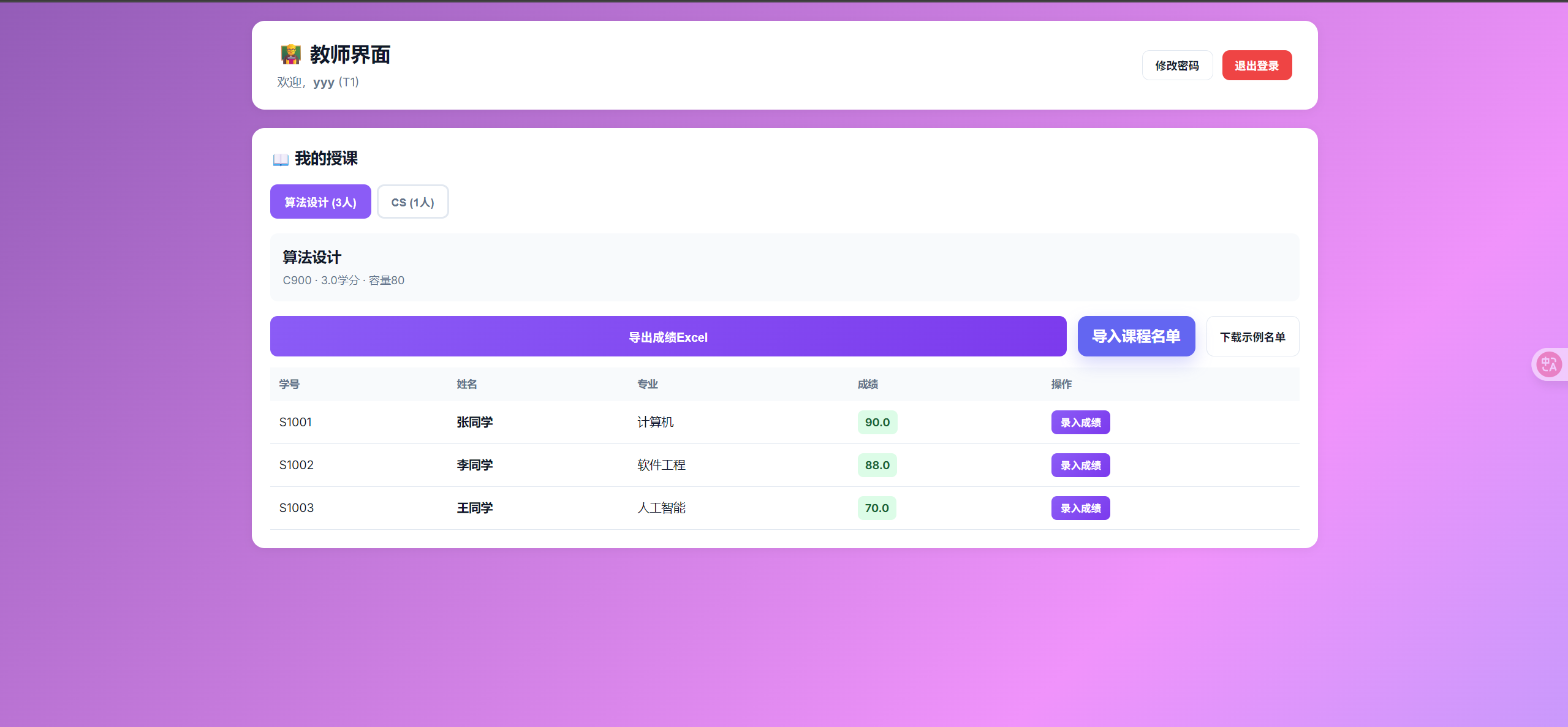
六、实验结果总结

运行效果：

学生端可正常选课/退选、查看成绩。



教师端能查看所授课程及名单，修改成绩即时生效。



管理端可通过 Excel 批量导入课程/学生/选课，导出成绩表，包含课程信息与成绩表，表头中文且命名按“课程-老师-时间”。





优点：

架构清晰，前后端分离；服务端分层（API/Service/DB）。

约束完善（唯一、外键、复合唯一），避免重复选课。

可选 SSH 隧道，适配云上私网数据库。

Excel 批处理提升数据运营效率。

不足与改进：

成绩审核与日志待完善（审计表）。

课程容量的并发控制可引入“扣减名额原子事务 + 队列”。

可以增加课程时间冲突检测与选课规则（先修课等）。

统一前端状态管理（Pinia）与接口错误提示更细致。

七、编程工作总结

**数据库设计经验**

先概念后逻辑：从 E-R（含 Admin/Student/Teacher/Course/Enrollment）到 DDL，逐步固化约束与索引。

约束前置：唯一键（username, student\_no, teacher\_no, course\_code）、复合唯一（student\_id, course\_id）防止重复选课，外键保证引用完整性，枚举/检查约束限定 status 与 role。

弹性字段：成绩允许为空以兼容“未评分”场景；教师关联可空以支持待分配课程；容量/学分设默认值减少数据录入成本。

索引实践：课程教师索引、选课学生/课程索引覆盖高频查询；复合唯一同时充当索引，兼顾一致性与性能。

数据导入健壮性：导入时先缓存现有键值映射（student\_no/teacher\_no/course\_code），幂等插入、跳过重复并汇总错误；缺失学生可按需创建以提高可用性。

**后端实现要点**

连接与安全：psycopg2 连接池 + 参数化查询防 SQL 注入；可选 SSH 隧道适配云上私网数据库。

分层解耦：蓝图按角色拆分（auth/student/teacher/admin），Service 层承载业务逻辑，DB 层只做数据访问。

会话与跨域：Flask-Session + SameSite=Lax Cookie，CORS 显式 origin + credentials，前端统一带凭证。

错误与权限：统一 json\_response/error\_response；@require\_auth 进行登录与角色校验；资源级校验（教师只能改自己课程的成绩，学生只能操作自己的选课）。

批处理与文件：pandas+openpyxl 读写 Excel；导出文件名包含课程名、授课老师与时间戳，表头中文；导入支持 courses/students/enrollments 多表。

**前端实现要点**

角色视图：根据用户角色呈现学生/教师/管理员界面；导航/登出统一。

数据拉取：封装 api() 统一 fetch，默认 credentials: 'include'；Promise.all 并行加载列表与统计。

UI 组织：列表/表单组件化，搜索与折叠可控；按钮态、错误提示、空态统一；统计卡片横向紧凑展示。

交互细节：删除/退课二次确认；导入/导出 loading/error 状态；成绩输入留空直接返回。

**工具与自动化**

Excel 工具链：pandas+openpyxl 处理导入/导出；示例生成脚本 generate\_sample\_excel.py 统一模板格式（courses/students/enrollments）。

启动与文档：PowerShell/命令行启动脚本，README 列出后端/前端启动步骤与 Excel 使用说明；.env 示例便于快速配置。

代码组织：前后端分离，Vite 开发热更新；后端模块化便于扩展（新增培养计划等功能时复用 Service/DB 层模式）。

**可选改进方向**

并发控制：选课容量扣减可加事务与行级锁或乐观并发控制，避免超额。

审计与日志：增加操作审计表，记录导入/导出、成绩修改、删除操作。

校验与规则：选课冲突检测、先修课程约束、成绩变更审批流。

前端状态：引入 Pinia/Redux 做全局状态与错误统一收敛，提升可维护性。

附：Excel 示例与导出

一键生成导入模板

cd /path/to/repository/backend/scripts

python generate\_sample\_excel.py   # 生成 sample\_import.xlsx

导入模板各表中文表头

课程(courses)：课程号, 课程名, 学分, 容量, 授课工号, 授课教师

学生(students)：学号, 姓名, 专业

选课(enrollments)：课程号, 学号, 成绩, 状态

成绩导出

管理员界面选择课程 → “导出成绩Excel”

文件命名：课程名称-授课老师-YYYYMMDD\_HHMMSS.xlsx

表头：课程号, 课程名, 学分, 授课教师, 学号, 姓名, 成绩, 选课状态实验