山东大学软件学院

Web技术 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202000300021 | 姓名：李泽宇 | | 班级：7班 |
| 实验题目：学生选课管理系统 | | | |
| 实验学时：16 | | 实验日期：2021年9月10日至12月21日 | |
| 实验目的：  熟练掌握和灵活应用网站的规划设计方面的理论知识和技能，规划和设计网站，使设计的网站能够对网页布局有充分的掌控能力，合理的选择颜色、字体、分辨率以及图片、声音、动画等多媒体元素，能最终实现网站的前端设计目标。  熟练掌握和灵活应用后端技术，使用java spring boot框架实现与前端的数据交互，实现对数据库的访问。 | | | |
| 硬件环境： PC、八核、64GB内存 | | | |
| 软件环境：  操作系统：（Windows）  开发工具：（IDEA、VS Code）  数据服务：（MySql-8）  应用服务：（Tomcat） | | | |
| 输出成果：  202000300021\_李泽宇\_实验1报告.docx  202000300021\_李泽宇\_实验1.zip，内容包括：  /CourseSelClient.zip 前端源码  /CourseSelServer.zip 后端源码  s202000300021\_total.jar 前后端集成产物  init.sql DB初始化脚本(后端源码CourseSelServer.zip中也有) | | | |
| 项目说明：  项目客户为老师，工期为一学期，计划投入1人两个月。  项目组共有1人，负责前后端开发、数据库设计等。 | | | |
| 需求分析 系统需要包括学生基本信息维护，学生选课，学生退课, 学生成绩查看, 生成可视化课表; 教师基本信息维护, 教师开课, 教师停开课程, 教师查看所开课程, 教师查看课程所含选课学生, 教师录入学生成绩功能.  系统应能支持500总用户量, 500人在线使用，页面最长响应时间为0.3秒。 | | | |
| 概要设计 分为用户管理、课程管理、选课进程管理、系统用户分类模块。  用户分为管理员、教师、学生三种角色。  功能简述如下：  1. 用户管理模块:  新用户的注册, 用户的登录, 用户信息的更新维护, 用户身份的判别  2. 课程管理模块:  课程与用户之间的对应关系, 课程信息的获取, 课程信息的修改.  3. 选课进程管理模块:  管理员控制选课系统的开放与关闭  4. 系统用户分类模块:  用户身份不同访问权限不同, 可访问的资源不同 系统管理 系统预先设定一个管理员账户, 账号为”admin”, 密码为 “123456”. 管理员登陆后可修改账号的密码. 管理员用户可以开启特定学期的选课功能, 也可以在一定的时间关闭选课系统. 同时, 管理员既可以以教师身份登录系统, 使用教师用户所拥有的全部功能, 又可以以学生身份登录, 使用学生用户所拥有的全部功能。  只有在管理原开启某一学期的选课功能后，教师用户才能开设这一学期的课程，学生用户才能在选课入口页面进入选课界面。 用户信息维护 系统管理员, 教师, 学生都属于用户. 用户信息包括真实姓名, 用户名, 性别, 大学, 电话号码, 登陆密码, 专业, 用户类型. 以上信息均为用户注册时所必须提供的. 用户名是一个用户的唯一标识, 用户名注册后不可再更改, 但其他信息可在用户登录后进行相关的修改更新. 课程设置 每一门课程包括以下信息: 课程序号(不同大学有不同的课程序号命名含义), 课程名, 开课学期, 开课教师, 开课时间, 开课地点, 学分, 学时, 课程性质, 所面向的专业, 选课的学生, 课程容量, 当前选课人数.  教师用户开设课程时也必须提供以上信息, 其中开课学期由系统管理员开启选课功能时提供, 教师用户不能自行开课学期.  教师开设课程后, 课程会在学生选课查看课程的界面搜索到此课程. 学生选课 系统管理员开启选课功能后, 学生用户可以进入到一个选课界面. 在此界面下学生可以查看当前学期所有开设的课程, 并且可以根据课程序号, 授课教师, 开课时间, 课程性质进行课程的查询. 当学生选择一门课程时, 此课程的相关信息会加入到学生已经选择的课程中, 并在学生的课表上相应的时间格子中添加此门课程. 当学生选择的课程的上课时间与某门已选课程冲突时, 会给予与”\*\*\*\*”课程时间冲突的提示; 当学生选择的课程的剩余容量为0时, 也会给出相关的提示; 当学生重复选择一门已经选择的课程时, 给出提示. 成绩录入 教师可在一个页面下查看自己开设的所有课程. 每个课程可以有相应的操作, 使得教师可以查看此门课程当前所有的选课学生的详细信息. 并可以在此界面录入学生的分数. 当选课功能正在开放时, 表示学生尚未选择完毕课程, 此时不允许教师录入成绩. 只有在系统管理员关闭选课功能后, 教师才可以录入学生的成绩信息. 录入信息后, 会生成一个饼状图, 来统计各个分段的学生人数, 方便教师查看本门课程选课学生的成绩分布. 学生成绩查看 学生直接进入某个页面即可查看自己课程的成绩情况, 学生可以获取到的信息有:  课程序号, 课程名, 授课学期, 授课教师, 课程学分, 课程性质, 自己的得分, 自己的排名, 此门课程所有学生的最高分和最低分. 特意加入满级最高最低分, 以便于学生更加客观地评价和了解自己的学习情况. 此处显示的课程仅为在教师已经录入学生成绩的课程, 即所有教师尚未录入成绩的课程是不会显示的. | | | |
| 系统架构 系统采用前后端分离的架构，前端使用Vue3 + Element Plus + axios + echarts框架，使用npm run build打包。  后端使用Spring Boot+ JPA + Spring Security框架。  目标用户环境为Edge, google浏览器版本以上，建议分辨率1080p。 | | | |
| 数据设计用户表user 包括以下字段：user\_id、real\_name、user\_name、gender、university、phone, password, major, person\_id, user\_type\_id, last\_login\_time, login\_count, desc, score  主键为user\_id。  user\_name字段有唯一约束。  gender、real\_name、user\_name、university、phone、password、major字段有非空约束。  user\_type\_id字段有外键约束，参照user\_type表id字段。  其中desc字段为用户的自我介绍，score字段是为了前后端传递数据时json对象的转化。Score字段实际不会存储一个固定值，只是为了前后端传递数据时及时获取用户某门课程的分数。 课程表course 包括以下字段：id、name、term、teacher、time、place、credit、classhour、property、major、students、capacity、currentnumber、sequence、remain、userid、score、max\_score、min\_score rank  主键为id  Id字段有唯一约束。  Name、term、teacher、time、place、credit、classhour、property、capacity、currentnumber字段有非空约束。  students字段中存储的为以“;”分割的学生用户的用户名。 课表表stoc 包括以下字段id、user\_id、term、courses。  主键为id  id字段有唯一约束。  user\_id、term字段有非空约束  此表的id为自动生成的自增主键，user\_id对应于用户表中的user\_name字段，term标识学期号，courses标识本学期与该用户有关的课程——若用户为学生，则此表的一项记录表示该学生在哪个学期选了哪些课程；若用户为教师，则此表的一项纪录表示该教师在该学期开设了哪些课程。courses字段中存储的为以“;”分割的课程id。 选课学期控制表sys 包括以下字段term、state。  term为主键  term、state字段有非空约束。  此表是系统管理用来开放和关闭选课学期、标识某一选课学期是否已经结束的表。  该表的每一项记录中，若state为1，表示term对应的选课学期已经结束——即term学期的选课阶段已经结束；若state为0则表示尚未选课结束。此表的所有记录中，只能有一项记录的state为0，其它记录的state均为1。因为特定的时间下，正在开放的选课学期只能存在一个。 用户类型表user\_type 包括以下字段id，name。  id为主键  id、name字段有非空约束。  此表的内容固定，记录的数目也固定。即：此表一共只有三项记录。id为0，对应的name为“USER\_ADMIN”，表示管理员用户；id为1，对应的name为“USER\_USER”，表示学生用户；id为2，对应的name为“USER\_TEACHER”，表示教师用户。  其中的id字段对应于用户表的user\_type 字段。 用户信息person 包括以下字段：person\_id，per\_name，per\_num，per\_type，per\_card，phone，email  person\_id为主键  此表其实并没有真正派上用场。编写此表的初衷本来是将其作为user表的外键，将user的详细信息放在该表中的，但是在后续编写代码的过程中，采用了直接将user的详细信息放在user表中。  故实际上，此表中没有任何信息，但此表可利于后续功能的扩展。 | | | |
| 功能接口用户登录 功能：  1. 对用户的输入进行检查，若为空给予提示。  2. 对登录用户的帐号、口令、身份进行认证。若身份、口令、密码三者均匹配，跳转到选课系统的用户主界面；否则给出用户错误警示。  如图片2，图片3，图片4，图片5。  地址：/api/auth/login  调用参数：  返回参数：  {  "username": "tom",  "password": "602bdc204140db016bee5374895e5568ce422fabe17e064061d80097"  "identity": "tea" // 或者 "stu"  }  {  "err": 0,  "access\_token": "e642824c3f8cf24ad09234ee7d3c76e2"  } 用户注册 功能：   1. 对用户填写信息的检查：不能有空内容、联系电话11位、口令一致性验证、学号已注册判定。 2. 注册成功跳转到登录界面 3. 向后台传递用户注册数据   地址：/api/auth/signup  调用参数：  返回参数：  {  "message": "User registered successfully" // 或"Error: Username is already taken!"  }  {    "username": username,         "password": password,         "realName": real\_name,         "gender": gender,         "university": university,         "phone": phone,         "major": major,         "role: role,  } 修改个人信息 功能：  对用户基本信息：性别，大学，专业进行修改  地址：/api/auth/update  请求参数：  {          username: 'your account number',          name: 'your name',          desc: 'your des',          university: 'your university',          major: 'your major',          gender: 'your gender',   }  返回参数：  {  "message": "success"  } 获取我的课程 请求地址：/api/course/mycourse  请求参数：  返回参数：  {    "data":  [{  "id": id,  "name": name,  "term": term,  "teacher": teacher,  "time": time,  "place": place,  "credit": credit,  "classhour": classHour,  "property": property,  },  …  {…}  ]  }  {    "username": username  } …………… 由于接口过多，此处不再一一列举。 | | | |
| 界面说明登录界面   图片1登陆界面    图片3空账号警示  图片2空身份警示    图片5登录信息错误    图片4空密码警示  相关输入警示和登录错误见图片，登录成功直接跳转到用户主界面。 注册界面   图片6注册界面    图片7未填充警示  图片8电话位数警示    图片10相异口令警示  图片9密码位数警示    图片11用户名重复错误  学号输入无要求，只要不跟已存在的注册学号相同。  表中所有内容必须填充。  电话号码要求11位。  注册成功后跳转到登录界面。 学生主界面   图片12学生主界面  主界面分为顶部导航栏和侧边导航栏。实现用户不同界面之间的跳转。  顶部导航栏点击时，刷新方式为整个页面刷新。  侧边导航栏点滴时，只刷新最右侧的组件。  在主界面用户可以实现向个人信息页面、我的成绩页面、选课入口页面、我的课表页面的跳转。 选课界面   图片13选课界面    图表14提示信息    图片15已选课程与退课  图片16教师开课界面  选课页面顶部为筛选条件，用户可根据其中某一个属性筛选课程，也可以将不同筛选条件进行组合来进行筛选。比如：用户可以根据课程名查询课程，根据课程名的查询方式为模糊查询；也可以查询课程性质为必修的，在某个星期某一节课开设的课程……  用户可以点击每一条记录最右侧的选课按钮来实现对课程的选择。还可点击绿色的已选课程按钮来查看我已经选择的课程。在已查看课程栏里，用户还可以通过点击退课按钮来退课。 教师主界面   图片16教师主界面    图片17教师开设课程界面    图片18教师查看开设课程及选课学生界面    图片19教师录入成绩及分数统计饼图 系统管理员界面   图片20系统管理原界面    图片21开放选课功能    图片22关闭选课功能  系统管理员的功能主要就是开启和关闭选课系统。 | | | |
| 安全设计 为抵御注入攻击，采用了参数化查询……等方式。  口令采用BCryptPasswordEncoder加密。  跨域请求的配置:    各个接口资源访问前的权限验证：   部署说明 环境需求：按照3000用户量估算，需要2台服务器，每台8核、64GB内存。初始存储64GB，每年扩充多少24GB。网络带宽1000MB。  部署方法：  0. 我已经在老师提供的远程sql数据库上部署完毕后端数据库了。即如下mysql地址：      老师只需要运行jar包，浏览器访问localhost:9091即可。但我依然提供了初始化的sql脚本。   1. 创建DB webteach，执行init.sql脚本。   此处使用的为老师提供的远程数据库和数据库连接。故DB不再需要创建。     1. 前端项目使用8080端口，后端项目使用9091端口。 2. 执行完init.sql脚本后，直接运行jar包，浏览器访问localhost:9091端口即可。 3. 已有三个内置账户“202000300021”、“T202000300021”、“admin”，分别为学生用户、教师用户、管理员用户，账户的密码均为“123456”。   若表结构被修改，表内容被篡改，可以重新执行本人提供的init.sql脚本，重新初始化。 | | | |
| 总结讨论 通过《学生选课管理系统》项目的开发，我掌握了spring boot和vue的基本开发方法和流程，了解了html5的标签使用方法，理解了DOM，BOM对象，熟悉了java script的基本用法。对前端的ajax异步请求有了一定理解。初步尝试了VUE组件间的传值情况，子组件可以传值给父组件，父组件也可以传值给子组件。在父子组件的传值中，父组件中值的更新会立即引发子组件中值的更新，但子组件中值的更新不会引发父组件的变化。  不足之处：对VUE的路由机制和spring boot的自动装配原理等概念还是不甚了解。  对这两个框架的理解还处于粗浅层面，只是会使用一些基本的开发功能。项目的并发性处理并没有做的很好，在高并发的情况下很可能会出现一些错误。对Spring Security的配置不了解。  下步工作：通过其他的web项目和实践操作来增进自己对vue和spring boot的理解。  可以通过查阅VUE和Spring Boot的官方文档来增进理解，并学习更多关于这两个框架的用法。 | | | |