



**QG工作室项目报告**

学 院 信息工程学院

专 业 电子信息类

班 级 电信4班

组 别 后台组

姓 名 颜钦煜

学 号 3123002217

2023年 月 日

广东工业大学计算机学院制

目录

[1. 项目简介 1](#_Toc30766)

[2. 设计思路 2](#_Toc24252)

[3. 详细功能设计 3](#_Toc20914)

[4. 程序测试 3](#_Toc7265)

[5. 项目亮点 3](#_Toc19276)

[6. 心得体会 3](#_Toc25747)

# 项目简介

QG在线学习平台旨在为学生和教师打造一个充满活力、灵活而互动的教育环境。对于学生而言，他们可以根据自身的兴趣和学习需求，轻松地在平台上浏览、选择来自各个教师开设的丰富多彩的选修课程。这些课程不仅涵盖了各种学科和领域，还提供了多样的学习资源和活动，如课程章节学习、在线答题等，让学生能够深度学习并扩展知识面。

对于教师而言，他们可以轻松地创建和管理自己的课程内容，灵活调整课程设置和教学方式，以满足不同学生的学习需求。通过平台提供的监控工具，教师能够实时跟踪学生的学习进度和表现，包括课程章节的得分情况、平均分统计，以及学生的学习记录和反馈。这为教师提供了宝贵的数据支持，帮助他们更好地了解学生的学习情况，因材施教，提供个性化的指导和支持。

总体而言，这一教育平台旨在提升教学效率和学习成效，为教育教学带来创新的解决方案。通过充分利用科技手段，构建起学生与教师之间紧密互动的学习生态，促进知识的传播和学术的交流，为广大学习者和教育工作者创造更加便捷、高效的学习与教学环境。

# 设计思路

本项目依托Javase,javaweb,mysql,tomcat等技术进行开发,整个项目分为前端，后台，与数据库交互三个部分。

1. 前端:使用element ui和Vue框架进行开发
2. 后台:基于baseServlet进行开发
3. 与数据库交互：基于JDBC技术和动态代理，反射等思想设计类MyBatis来实现与数据库交互

技术选型：

前端：HTML，CSS，JavaScript，Vue，Element Ui

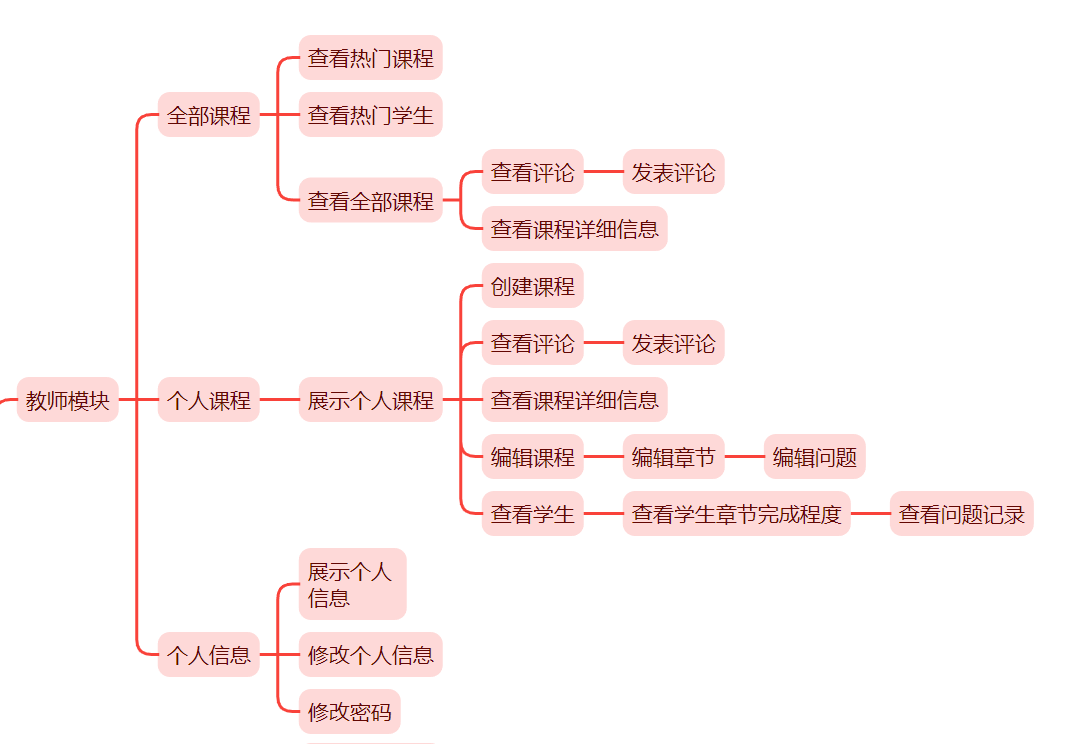
后端：Javase，MySQL，Maven, Servlet,

前后端交互：AJAX。

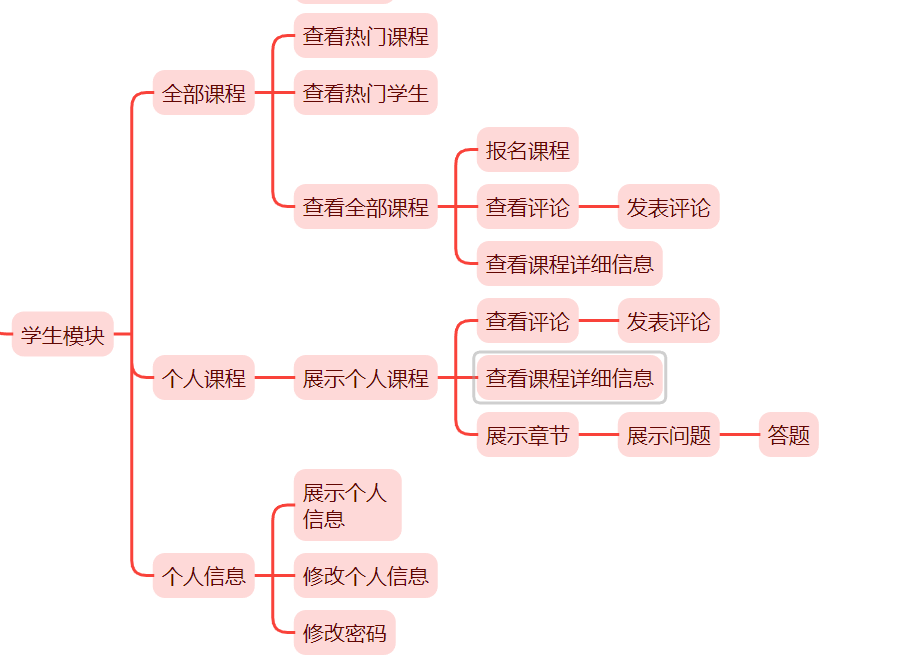
项目要求:

1. 登录与注册
2. 查看与管理个人信息
3. 教师创建，编辑个人课程，章节和题目。
4. 教师查看课程学生的学习情况，平均成绩，课程评论区等。
5. 学生报名参加课程，进行课程学习，查看课程的学习情况，答题记录等
6. 用户可以查看热门的课程排行榜，和学习情况较好的用户排行榜。
7. 支持验证登录，对信息敏感数据加密传播
8. 前后端都做好正则表达式校验
9. 使用JUnit进行单元测试

教师模块设计思路：



学生模块设计思路



# 详细功能设计

1. 数据库表格设计：

数据库表格主要分为三大类，

1. 用户类:学生和老师，用户类主要是储存用户的个人信息，要点是密码要使用加密，做好用户账号安全。
2. 课程类:课程，章节，问题，课程详细信息，评论。课程类要注重各个表单之间的联系，课程要带有课程详细信息的Id,评论要带有所属课程的id，章节要带有课程id，问题到带有章节id，在进行查询时便依据id进行查询
3. 学习类:注册课程的学生，当前章节，当前问题，学习类是学生报名课程后生成的，分别对应课程，章节，问题，要主要这三个表格与课程的对应关系，当课程的一些数据修改或课程直接删除时，对应的学习类也要进行改变。

Student:

1. Id
2. Username(姓名)
3. Password(密码)
4. PersonInformation(个人介绍)
5. StudentNumber(学号)
6. Grade(年级)

Teacher:

1. Id
2. Username(姓名)
3. Password(密码)
4. PersonInformation(个人介绍)
5. qq(QQ号)
6. Email(邮箱地址)

Lesson:

1. Id
2. Name(课程名)
3. TeacherName(教师名)
4. InformationId(课程详细信息Id)
5. CurrentEnrollment(当前报名人数)
6. RegistrationPopulation(总报名人数)

Chapter:

1. Id
2. Name(章节名)
3. LessonId(章节所属课程的id)
4. Information(章节信息)
5. ImagePath(图片路径)

Question:

1. Id
2. Name(问题名)
3. ChapterId(问题所属章节的id)
4. Information(问题信息)
5. ImagePath(图片路径)
6. QuestionType(问题类型)
7. RightAnswer(正确答案)
8. Score(分数)

LessonInformation:

1. id
2. BeginTime(课程开始时间)
3. Information(信息)
4. EndTime(课程结束时间)
5. Name(课程名)
6. RegistrationPopulation(总报名人数)

Discussion:

1. id
2. DiscussName(评论人姓名)
3. Information(评论信息)
4. LessonId(评论所属课程Id)
5. Time(评论时间)

RegisterStudent:

1. id
2. StudentName(学生姓名)
3. LessonId(课程Id)
4. GetScore(在该课程的得分)

CurrentChapter:

1. id
2. StudentName(学生姓名)
3. ChapterId(章节Id)
4. ChapterFinish(章节完成程度)
5. Information(章节信息)
6. LessonId(课程Id)
7. GetScore(章节得分)
8. ImagePath(图片路径)

CurrentQuestion:

1. id
2. StudentName(学生姓名)
3. QuestionFinish(问题完成程度)
4. Answer(学生提交的答案)
5. Information(问题信息)
6. QuestionId(问题Id)
7. GetScore(问题得分)
8. ImagePath(图片路径)
9. ChapterId(章节Id)

2.Filter:

LoginFilter是一个Java Servlet过滤器，用于对Web应用程序中的请求进行过滤，以确保只有经过身份验证的用户才能访问受保护的资源。该过滤器实现了Filter接口，并覆盖了init、doFilter和destroy方法。

在doFilter方法中，首先获取了HTTP请求对象，并提取了请求的URL。然后，它定义了两个字符串数组urls，分别存储了允许直接访问的URL模式。如果请求的URL匹配到了这些允许访问的模式之一，则直接放行，不进行进一步的处理。

如果请求的URL不属于允许直接访问的模式之一，接着会检查是否是需要进行身份验证的URL模式。如果是需要身份验证的URL模式，则会检查请求中是否包含Authorization头或Authorization参数，如果包含了有效的Authorization信息，使用JwtUtils工具类解析JWT令牌。如果JWT解析成功且包含有效的用户名信息，则允许请求通过，否则拦截请求。并返回未授权的响应。

该过滤器的作用是确保对于需要身份验证的资 源，只有携带有效的身份验证信息的用户才能访 问，以提高Web应用程序的安全性。

3.登录与注册:

LoginServlet:

用户请求参数解析:使用ReadJSONUtils.parseJsonToClass(req, User.class)方法将HTTP请求中的JSON数据解析为User对象。

数据验证：使用Regex.validateName()和Regex.validatePassword()方法验证用户输入的用户名和密码是否符合规则。

业务逻辑处理：

调用PersonService接口的实现类PersonServiceImpl的方法，根据用户提供的用户名和角色类型查询数据库中的用户信息。

如果查询到了用户信息，并且用户输入的密码与数据库中存储的密码匹配，则执行以下操作：将用户对象的密码设置为数据库中存储的密码。

生成JWT令牌，将用户名和角色信息添加到JWT声明中。

创建JSON对象，将用户对象和JWT令牌添加到其中。

输出登录成功的消息和JSON响应。

如果未查询到用户信息或密码不匹配，则返回HTTP 401 Unauthorized响应，并向客户端发送错误消息。

LoginServlet处理了登录请求，包括验证用户身份、生成JWT令牌以及返回相应的JSON数据或错误消息。

RegisterServlet:

请求参数解析：使用JSONObject.parseObject() 方法将HTTP请求中的JSON数据解析为 Student对象。

数据验证：使用Regex.validateName()等方法验证用户输入的用户名、密码和学号是否符合规则。

业务逻辑处理：

调用PersonServiceImpl类的实例方法，该类实现了PersonService接口，用于处理与用户相关的业务逻辑。

调用getStudentBtNumber()方法，根据学号查询数据库中是否已存在相同学号的学生信息。如果存在相同学号的学生信息，则向客户端发送"Error"消息。

如果不存在相同学号的学生信息，则调用addPerson()方法将新的学生信息添加到数据库中，并向客户端发送添加结果的消息。

RegisterServlet处理了注册请求，包括验证用户输入的信息、检查学号是否已存在以及将新的学生信息添加到数据库中

4. 学生界面及功能设计：

首先，界面通过侧边栏分为三个主要部分，这种分组设计有助于用户快速导航和定位所需功能，提升了用户的操作便捷性。

1.全部课程界面：这个界面是学生浏览和选择课程的主要入口。在课程界面中，一个表格用于展示所有可选课程，表格中包括了课程名、授课老师以及当前报名人数等基本信息。通过一个按钮，用户可以查看每门课程的详细信息，包括开课时间、结课时间、总人数和课程介绍等。帮助学生更好地了解课程内容和安排.另外，还有一个按钮用于展示该课程的评论区，评论内容包括评论时间、评论人姓名以及评论内容，学生可以在这里看到其他同学的评价，从而更好地了解课程质量和教学效果，并且可以自己发表评论，分享自己的学习体验。最后，针对每门课程都设有报名按钮，学生可以通过点击按钮报名感兴趣的课程，报名成功后，课程将会出现在个人课程处，方便学生查看和学习。

热门课程：热门课程是一个重要的导航入口，它帮助学生快速找到受欢迎和热门的课程。在界面中设有热门课程按钮，用户点击后即可展示当前全部课程中的热门课程。这些课程通常是当前报名人数最多的课程，通过这种方式，学生可以更快速地了解到其他同学普遍喜爱的课程，提高了选课效率。

卷王排行：卷王排行是一个有趣而又有竞争性的功能模块。在界面中设有卷王排行按钮，用户点击后可以展示所有学生中的卷王，其标准是学生报名课程的得分之和最大的学生。这种设计激发了学生的竞争欲望，鼓励他们积极参与课程学习和考试，从而提高学习积极性和成绩水平。同时，这也是一个展示学生学习成绩和能力的平台，促进了学生之间的交流和分享。

2.个人课程界面:学生能够直观地浏览已报名的课程，并轻松获取详细信息。这个表格不仅提供了课程的基本信息，还让学生能够快速访问讨论区，以便与同学和老师交流。当学生确定要学习某门课程时，只需点击相应的按钮，系统将立即呈现出该课程的所有章节。在这里，学生可以按照自己的节奏逐一学习每个章节，并且随时了解自己的学习进度。此外，学生还可以在学习过程中进行答题练习，通过点击答题按钮，系统将呈现出当前章节的所有问题。这些问题可能包含文字描述或图片，学生需要根据自己的理解和知识来回答。答题完成后，系统会即时给出反馈和得分，帮助学生更好地掌握课程内容。这样的设计不仅方便了学生管理和学习课程，还能够提供个性化的学习体验，帮助他们更高效地掌握知识。

3.个人信息:

此界面功能有展示学生个人信息(姓名,个人介 绍,学号,年级),设有两个按钮，一个按钮可以修改 个人信息，另一个按钮可以修改密码

1. 教师界面及功能设计:

1.全部课程界面：在课程界面中，一个表格用于展示所有可选课程，表格中包括了课程名、授课老师以及当前报名人数等基本信息。通过一个按钮，用户可以查看每门课程的详细信息，包括开课时间、结课时间、总人数和课程介绍等。帮助教师更好地了解课程内容和安排.另外，还有一个按钮用于展示该课程的评论区，评论内容包括评论时间、评论人姓名以及评论内容，教师可以在这里看到同学们的评价，从而更好地了解课程质量和教学效果，并且可以自己发表评论。

热门课程：热门课程是一个重要的导航入口，它帮助学生快速找到受欢迎和热门的课程。在界面中设有热门课程按钮，用户点击后即可展示当前全部课程中的热门课程。这些课程通常是当前报名人数最多的课程，通过这种方式，教师可以更快速地了解到同学们普遍喜爱的课程。

卷王排行：卷王排行是一个有趣而又有竞争性的功能模块。在界面中设有卷王排行按钮，用户点击后可以展示所有学生中的卷王，其标准是学生报名课程的得分之和最大的学生。这种设计激发了学生的竞争欲望，鼓励他们积极参与课程学习和考试，从而提高学习积极性和成绩水平。同时，这也是一个展示学生学习成绩和能力的平台，促进了学生之间的交流和分享。

1. 个人课程界面：

教师能够轻松管理他们所负责的课程。这个界面提供了一个清晰的表格，列出了教师所属的所有课程，让他们能够一目了然地查看和处理每门课程。通过这个表格，教师可以点击相应的课程，以查看更详细的信息，包括课程介绍、目标、教学大纲等。同时，他们也能够进入评论区，与学生进行互动，了解他们对课程的反馈和建议。

1. 另外，教师还能够深入了解学生的学习情况。通过个人课程界面，教师可以快速查看当前课程已报名学生的学习进度。这包括每个学生在每个章节的完成情况、得分情况以及答题情况。这种即时的反馈让教师能够更好地了解学生的学习状态，及时调整教学策略，帮助学生更好地掌握课程内容。
2. 除了查看学生的学习情况，教师还可以对课程进行各种编辑操作。他们可以添加新的章节，以更好地组织课程内容；他们也可以为章节添加问题，丰富课程的互动性和趣味性。而且，教师还有权利在问题中添加图片，以更生动地呈现知识点，提升学习效果。此外，教师还可以删除不再需要的章节或问题，以保持课程内容的整洁和时效性。
3. 值得一提的是，教师在个人课程界面还可以新增全新的课程，灵活地扩展自己的教学范围。他们可以根据需要添加新的章节和问题，定制专属于自己的教学内容，为学生提供更多元化的学习选择。

3.个人信息：

此界面功能有展示教师个人信息(姓名,个人介 绍,qq号,邮箱号),设有两个按钮，一个按钮可以 修改个人信息，另一个按钮可以修改密码

# 4.程序测试

测试是开发过程的必经之路，一个好的程序离不开成千上万的测试，以下为本程序测试的标准及期望。

1. **功能测试**

功能覆盖率：测试功能覆盖全面，基本覆盖了所有功能点。

功能可用性：功能实用性高，能较好完成用户需求。

数据完整性：保证数据完整存储，处理过程中数据出现不会流失或者改变等问题。

1. **易用性测试**

操作简便性：用户可以快速找到所需功能，操作简单明了。

可理解性：用户能够理解每个操作的功能和意义，操作功能简单易懂。

易学性：新用户可以迅速理解该系统的功能和使用方法。

反馈及使用：用户能收获良好的用户体验。

1. **安全测试**

SQL注入攻击：防止用户输入恶意SQL语句对数据库进行攻击，

# 5.项目亮点

1. 手写MyBatis框架：

在数据库交互中，使用jdbc时，我们需要手动建立数据库链接、编码 SQL 语句、执行数据库操作、自己封装返回结果等，这些过程及其繁琐，尤其是在开发这样的javaweb项目时，数据库中表格数量增多，同时项目有着大量对于数据库的改查操作，再继续使用jdbc开发，会使开发过程繁杂且难受，所以在本项目中，我使用了自己编写的mybatis框架来简化开发。这里将简单讲述实现过程

一:解析mapper文件:

使用mapperConfig文件储存数据库连接信息和mapper文件路径，在mapper文件中储存具体的sql文本

我使用dom4j和xpath来实现解析mapper文件，通过获取element来得到文件信息

在XMLConfigBuilder中，先解析数据库连接池配置信息，构建数据库连接池。随后使用XMLMapperBuilder解析Mapper文件。

每个sql文本通过<select>标签确定sql语句的类型，parameterType确定参数类型，，resultType确定返回结果类型。

对每条sql语句创建XMLStatementBuilder处理，创建SqlSourceBuilder解析sql语句，使用GenericTokenParser将sql语句中{x}替换为?，将括号内的字符串构建参数映射，先根据parameterType是否在类型处理注册机中，如果在则参数类型设为parameterType，否则解析parameterType获取metaClass，使用反射获取参数类型。

将其加入参数映射。

使用MapperBuilderAssistant 构建器助手创建 MappedStatement 映射语句类，在这个类中封装了入参和出参的映射、以及把这些配置信息写入到 Configuration 配 置项中，还有对映射器的 ID 进行命名空间前缀处理、

二：建立数据库连接池

在项目中，包括两部分分为无池化的 UnpooledDataSource 实现类和有池化的 PooledDataSource 实现类，池化的实PooledDataSource 以对无池化的 UnpooledDataSource 进行扩展处理。把创建出来的链接保存到内存中，记录为空闲链接和活跃链接，在不同的阶段进行使用

使用PooledConnection ，通过invoke方法的反射调用，对关闭的链接进行回收处理，并使用 notifyAll 通知正在等待链接的用户进行抢链接。用户想要获取链接的时候，则会从连接池中进行获取，同时判断是否有空闲链接、最大活跃链接多少，以及是否需要等待处理或是最终抛出异常

三.通过反射完成一个对象的属性的反射填充

在设置参数以及处理返回结果时，我们如果使用硬编码设置，会使得我们的代码繁杂又不具有通用性，所有这里使用反射来解耦对象，

对于一个对象来说，它所包括的有对象的构造函数、对象的属性、对象的方法。而对象的方法因为都是获取和设置值的操作，所以基本都是get、set处理，所以需要把这些方法在对象拆解的过程中需要摘取出来进行保存。

以 Reflector 反射器类处理对象类中的 get/set 属性，包装为可调用的 Invoker 反射类，这样在对 get/set 方法反射调用的时候，使用方法名称获取对应的 Invoker 即可 getGetInvoker(String propertyName)。

有了反射器的处理，之后就是对原对象的包装了，由 SystemMetaObject 提供创建 MetaObject 元对象的方法，将我们需要处理的对象进行拆解和 ObjectWrapper 对象包装处理。因为一个对象的类型还需要进行一条细节的处理，以及属性信息的拆解， 这样一个类中的关联类的属性，则需要进行递归的方式拆解处理后，才能设置和获取属性值。

最终在 Mybatis 其他的地方就可以有需要属性值设定时，就可以使用到反射工具包进行处理了。

四．映射器:

MapperMethod:判断方法的类型（如Select,Update）,并获取方法的参数

MapperProxy: 负责实现 InvocationHandler 接口的 invoke 方法，最终所有的实际调用都会调用到这个方法包装的逻辑。

MapperProxyFactory 是对 MapperProxy 的包装，对外提供实例化对象的操作。省去了我们创建MapperProxy时编写 Proxy.newProxyInstance 进行处理。

MapperRegistry 提供包路径的扫描和映射器代理类注册机服务，完成接口对象的代理类注册处理，

五．SqlsessionFactory和Sqlsession:

构建sqlsessonFactory并创建sqlsession，sqlsession

中存有配置Configuration和执行器executor，

Configuration:配置类存有

1.环境(Environment)，

2.映射注册机( MapperRegistry)，

3.映射语句(mappedStatements),

4.类型别名注册机(typeAliasRegistry),

5.语言驱动注册机(languageRegistry),

6.类型处理器注册机(typeHandlerRegistry)

7.对象工厂和对象包装器工厂

(objectWrapperFactory objectFactory)

8.加载资源(loadedResources)

执行器executor:

执行器是sql命令的直接执行者，sqlsession的方法如selectOne就是调用执行器executor的doQuary方法执行sql语句。doQuary方法中调用了jdbc类实现对数据库的增删改查，

六．参数设置并执行sql语句：

调用DefaultParameterHandler中setParameters方法，方法会先获取ParameterMapperings，遍历集合，给sql语句中每个?设置参数，如果传入参数类型是存在与typeHandlerRegistry中,则直接赋值。如果不存在则通过 MetaObject.getValue 反射取得值设置进去。之后获取参数在数据库中的jdbc类型，执行typeHandler.setParameter方法设置参数，设置好参数后便执行sql语句

七．返回结果集resultset处理:

先对resultset进行包装，获取metaData，遍历metaData获取columnName，jdbcType，.className。在从mapperStatement中获取resultMap,根据resultMap中的resultType创建返回对象，如果返回对象类型是int,string,long。这直接给返回对象赋值。否则创建metaObject，根据resultMap获取未映射的属性名集合，遍历集合，如果属性名在metaObject存在，这将其赋值。将赋值完后的meta Object对象储存在multipleResults中，最后如果multipleResults长度为1，则返回第一个，否则返回集合。

1. 使用jwt令牌储存用户信息，在登录处理中，成功查询到用户信息后，会生成jwt令牌并传回前端，前端将其储存到本地，以后每次请求都会在filter检查jwt令牌
2. 使用MD5+盐的形式处理密码，并储存到数据库中，提高安全性。
3. 使用阿里云oss储存图片,对于课程和章节的图片，储存在阿里云oss中，使获取与储存更加便捷

5.后台代码传回前端时，会根据返回结果使用element ui的$message产生弹窗提醒请求结果

6.在前后端都使用正则表达式校验登录与注册数据，如密码在十位以内，且只能是数字或英文字母。

7.前后端数据传输都严格使用Json格式，前端传输的数据都JSONObject.parseObject转化为类。后台传输区前端的数据使用JSON.toJSONString方法转化为JSON字符串。

8.编写了Utils工具类:减少重复代码，

1.TimeUtils时间工具类:

获取现在时间并生成为格式为 年- 月-日 的字符串；接收开始时间与 结束时间，判断现在时间是否在 区间之内

2.SaltMD5Util密码加密工具类：

1.使用generateSaltPassword 生成经过 盐加密的密码。返回一个包含盐和 密 码哈希值的字符串。

实现：首先生成一个随机的16位盐， 然后将盐和密码拼接起来进行 MD5 哈希处理，最后将盐和哈希 后的密码 混合在一起返回。这样生 成的字符串 包含了盐和密码的哈 希值，提高了密 码的安全性。

2.使用verifySaltPassword 验证经过盐 加密的密码是否与明文密码匹 配。 如果密码匹配，则返回 true； 否则返回 false。

实现：从经过盐加密的密码字符串 中提取出盐和哈希后的密码，然后 使用相同的盐和明文密码进行哈希 处理，最后将哈希后的结果与提取 出的密码哈希值进行比较，以验证 密码是否匹配。

使用md5Hex 对字符串进行 MD5 哈希处理。 返回经过 MD5 哈希处 理后的字符串。

实现：使用Java 的中 的 MessageDigest 类 获取 MD5 摘要实例，然后对输入字 符串进行 字节流的 MD5 哈希处理， 最后将 处理后的字节数组转换成字符 串 返回。

这些方法结合了盐加密和哈希处理 的技术，提高了密码存储的安全性， 并且通过验证方法可以有效地验证 用户输入的密码是否正确。

3.JwtUtilsJwt令牌：

1.使用HMAC-SHA256 签名密钥。 生 成 Jwt令牌，

2.解析Jwt令牌

该工具类的作用是提供了生成和解析 JWT 令 牌的方法，方便在应用程序中进 行身份验证和授权操作

9.使用junit进行测试:

在项目开发中，无论是数据库的命令调试， 还是工具类的调用，我都会使用junit进行 测试，提高开发效率

**6.项目难点：**

在项目的开发过程中，产生了许多难点，克服与解决难点的过程，本身就是一大收获，

1. 前端界面样式错误：在前端界面的搭建中，经常因为组件放置错误，使得前端界面直接无法显示。要仔细查看组件搭配，阅读必要前端知识。
2. 前后端请求发送数据产生错误：在发生数据时，因为没有设置好输送格式，使得数据接收发生遗漏，要严格按照json格式发送，做好前后端检查
3. 后台处理事务逻辑输错，发生死循环：要仔细检查事务逻辑，使用junit进行测试，使用debug工具进行追踪，查清问题
4. Sql语句编写错误：sql语句经常因为与数据库类名不匹配，产生错误，要在使用前使用junit进行测试，发现问题并解决。

# 7.心得体会：

这次的项目，根据项目要求文档，使用javase,javaweb等技术成功地构建了一个功能齐全、界面美观的javaweb项目。在这个过程中，我收获了许多宝贵的经验和感悟。以下是我对这次制作项目的心得总结：

一、对于项目的制作:

（1）明确项目需求和目标：在启动项目之前，必须清晰地定义项目的需求和目标。这包括确定项目的功能、设计要求、交付时间和期望效果。只有明确了这些方面，才能有针对性地进行后续的开发工作。

（2）设计项目架构：在开始编码之前，应该先设计好项目的整体架构。这包括设计数据表格结构、后台处理逻辑、前端界面设计以及前后端交互的请求处理逻辑。良好的架构设计能够为后续的开发工作提供清晰的指导和框架。

（3）选择合适的工具和技术：在项目开发过程中，选择适当的工具和技术可以提高开发效率和项目质量。例如，可以使用阿里云OSS来存储图片等静态资源，同时使用JSON作为前后端交互的数据格式。选择合适的工具和技术有助于简化开发流程并提升项目的可维护性和扩展性。

1. 测试和优化项目性能：项目开发完成后，必须进行全面的测试和性能优化工作。这包括测试项目在不同浏览器和设备上的兼容性，以及优化项目的性能和加载速度。通过测试和优化，可以确保项目的稳定性和良好的性能表现，提升用户体验并减少潜在的问题和风险。

二、对于技术方面的感悟：

对于我来说，掌握了诸如Maven、Tomcat、Git、Element UI、Vue框架和Axios等关键技术，对我来说是开发中的一大利器。首先，Maven作为项目管理工具，简化了技术引入的流程，让我能够更轻松地集成各种技术和库，极大地提升了开发效率。Tomcat作为一个流行的Servlet容器和Web服务器，为我提供了一个可靠的运行环境，使得我的项目能够高效地运行和部署。

另外，Git作为版本控制系统，不仅让我的代码管理更加规范和高效，还让团队协作变得更加顺畅。Element UI和Vue框架作为前端开发的利器，极大地简化了前端界面设计的过程，让我能够更专注于业务逻辑的实现，同时提升了用户体验。通过这些工具，我能够快速构建出美观、交互丰富的前端界面，为用户提供更加友好的使用体验。

而Axios作为一个基于Promise的HTTP客户端，为我的前端开发提供了强大的异步请求发送能力。它简洁而灵活的API让我能够轻松地发送各种类型的请求，并对返回的数据进行处理，极大地提升了前端与后端数据交互的效率和可靠性。

我熟练掌握这些关键技术，不仅让我的开发工作事半功倍，还让我能够更加专注于业务逻辑的实现和用户体验的优化，为项目的顺利进行和用户的满意度提升提供了坚实的技术支持。

关于这次的项目，

1. 前端：我使用了element ui,Vue框架，axios等技术来制作前端界面，界面设计使用element ui内置的组件，数据和逻辑处理使用了Vue框架，使用axios发送前端异步请求。在开发中我对前端的知识有了充分了解，也意识到了前端开发中逻辑处理的重要性。
2. 后端：这次项目中我使用mysql数据库存储数据，按照Servlet->Service->mapper的流程处理请求，在前端请求发送到后台时，根据请求路径进入Servlet，调用Service处理数据，使用基于jdbc开发的Mybatis框架进行与数据库交互，最后Servlet根据返回结果给前端发送数据

（3）项目的调试与优化：在项目制作的过程中，常常遇到了很多问题和挑战。有时是一个小小的bug让我纠结半天，有时是一个复杂的功能让我束手无策。但正是这些困难和挑战，让我更加深入地了解了项目开发的细节和技巧。在调试过程中，运用浏览器的开发者工具来查看和修改页面的元素和样式，使用控制台来查看和打印变量的值。运用这些工具让我更快地定位和解决问题。在优化方面，我注重了代码的可读性和可维护性。我尽量让代码结构清晰、逻辑简单，避免使用过于复杂的算法和技巧。

关于代码编写，

（1）代码注释的重要性：

1.提高代码可读性：通过添加注释，可以帮助其他开发人员更容易地理解代码的意图和逻辑。清晰的注释可以使代码更易于阅读和理解，缩短新人入门的时间。

2.方便代码维护：在代码中添加注释可以帮助开发人员快速定位和理解特定功能或模块的实现方式，从而更容易进行代码的维护和修改。

3.减少错误和bug：通过注释可以提醒开发人员注意某些特殊情况或潜在的问题，有助于减少代码中的错误和bug，提高代码的质量和稳定性。

4.遵循最佳实践：良好的注释规范是编程的最佳实践之一，有助于形成团队统一的编码风格和规范，提高代码的一致性和可维护性。

（2）模块化的重要性：

1.提高代码复用性：通过模块化的设计，可以将代码分解成独立的模块，提高代码的复用性和可扩展性。可以更方便地在不同项目中重复使用模块，减少重复编写相似代码的工作量。

2.降低耦合度：模块化设计可以降低代码之间的耦合度，使各个模块之间的依赖关系更清晰和简单。这样可以减少代码的修改范围，提高代码的灵活性和可维护性。

3.提高开发效率：通过模块化的设计，可以使团队成员更好地分工合作，各自负责不同模块的开发和维护。这样可以提高开发效率，加快项目的进度。

4.易于测试和调试：模块化的设计使代码更易于测试和调试，可以单独测试和调试每个模块，减少错误的影响范围，提高代码的质量和稳定性。

5.支持扩展和升级：通过模块化的设计，可以更容易地扩展和升级项目，添加新功能或替换旧模块。这样可以使项目更具有灵活性和可扩展性，满足不断变化的需求。

关于项目调试，

在项目开发中，debug是极其重要的一步，整个项目跨度巨大，从前端一直数据库，逻辑复杂，难免会产生bug，这时，使用debug工具一步步追踪，解决bug就显得尤为关键。Debug的过程也令我收获颇丰，我通过追踪了解了数据传递的过程，复杂逻辑的执行过程。这些都对我理清项目起到了巨大的作用。

三、制作本项目的体会：

这次项目制作时间较紧张，且不能使用一些框架，代码繁杂，制作的过程难免出现较多的bug，如：页面设计的时候样式位置容易因为一个小错误而出现整体错乱，sql语句编写过程中逗号或括号写错导致错误。在修改数据库数据时没有考虑到各个表之间的联系。

因此，程序阶段性测试的重要性还是尤为重要的，通过阶段性测试，可以及早发现和识别软件中存在的问题和缺陷，有助于及时修复，避免问题在后续阶段扩大和影响整体开发进度。也有助于确保软件的质量和稳定性，通过对每个阶段的功能和性能进行测试，可以提前发现并解决潜在问题，从而提高项目的质量和可靠性。并且我觉得在未来团队开发中，阶段性测试可以帮助验证软件的需求和设计是否符合实际需求，是否满足用户的期望。通过测试可以确保软件按照设计和规格要求进行开发。

总的来说，开发qg在线学习平台是一个很好的技术实践机会，收获了很多宝贵的经验和感悟。通过这次开发，我不仅掌握了一些技术和工具，还初步有了制作一个大型项目的经验，对我日后的后台学习很有帮助。