**武汉大学计算机学院**

**本科生课程设计报告**

**互动教育类型Web项目——LearnHub平台总体设计与实现**

专 业 名 称 ：软件工程

课 程 名 称 ：Web程序设计与实验

团 队 名 称 ：LearnHub团队

指 导 教 师 ：韩波 教授

团 队 成 员 一： 郭志昊（2016302580040）

团 队 成 员 二： 张凯博（2016302580244）

团 队 成 员 三： 朱文炜（2016302580317）

团 队 成 员 四： 李汉杨（2016302580246）

团 队 成 员 五：陈毅（2016302580267）

团 队 成 员 六：周彪（2016302580062）

团 队 成 员 七： 谢志超（2016302580319）

团 队 成 员 八： 李朝钦（2015302540098）

团 队 成 员 九： 黄伟康（2016302580111）

团 队 成 员 十： 吕冠晖（2016302580266）

团 队 成 员 十一：刘湘（2016302580268）

团 队 成 员 十二：薛尊尧（2016302580245）

二○一九年六月

**郑 重 声 明**

本团队呈交的设计报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本设计报告不包含他人享有著作权的内容。对本设计报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本设计报告的知识产权归属于培养单位。

团队成员签名： 日期：

摘 要

本实验目的是实现Web项目——LearnHub互动教育平台。此平台针对当前教育社交类网站的市场空白，将视频课程、问答、博客等多种功能进行整合。力求将平台打造成集慕课、CSDN、知乎等网站优点于一体的交互社交类教育网站。

本Web项目后端运用SSM框架与MySQL数据库结合。前端通过HTML+jsp+JS实现，并运用了流行的Bootstrap前端框架。

**关键词：**教育平台； Web程序设计

**目 录**

**1** **软件设计阶段**

1.1 可行性分析 ……………………………………………………………………5

1.2 用户故事定义………………………………………………………………… 5

1.3 原型图绘制……………………………………………………………………10

**2 编码实现阶段**

2.1 项目结构………………………………………………………………………13

2.2 实现方案………………………………………………………………………14

2.3 数据库ER图设计………………………………………………………………17

**3 项目成品展示** ………………………………………………………………23

**1 软件设计阶段**

**1.1** **可行性分析及需求分析**

本部分仅就市场可行性进行讨论。

根据调研，当前市场上的教育类网站分为几个类型：一是以结构化的知识体系为主，提供各类课程视频，分章节展示，如慕课。二是以用户上传各类学习资源为主，用户间可进行分享交流，如CSDN、知乎。

这些类型的网站各自都有各自的局限性。

就功能设计上讲，他们各自有各自独特的功能，慕课以视频课程为主，CSDN以文章博客为主，知乎以问答为主。这些功能都是很有特色且优秀的设计，但为什么不能有一个网站兼具上述各种功能呢？

就需求分析上讲，慕课的功能偏向死板、枯燥，缺少用户之间的交互，使得用户总体的学习热情和使用度并不高。知乎和CSDN用户交流很是活跃，但缺少权威化的、系统化的知识结构和学习资源。

为弥补上述的局限性，LearnHub力求做到功能和需求的整合，有很好的潜在用户基础和市场前景。

**1.2 用户故事定义**

根据上述的需求分析，我们定义出了以下用户故事以及其对应的验收条件。

作为游客，可以注册成为用户，以便获取使用用户的权限 1.输入符合要求的用户名和密码以及验证码。预期结果：注册成功并转到已登录的首页。2.输入不符合要求的密码，符合要求的用户名及验证码。预期结果：提示密码不符合要求。3.输入符合要求的用户名和密码，错误的验证码。预期结果：提示验证码错误。4.输入重复的用户名，符合要求的密码和验证码。预期结果：提示用户名已存在。

作为用户，可以登录系统，以便使用用户的功能 1.输入正确的用户名密码以及验证码。预期结果：登录成功并转到已登录的首页。2.输入正确的用户名和密码，错误的验证码。预期结果：提示验证码错误。3.输入错误的用户名或密码，正确的验证码。预期结果：提示用户名或密码错误。

作为用户，可以注销登录，以便重新登录 1.已登录用户点击注销按钮。预期结果：退出登录并转到登录界面。2.未登录用户点击注销按钮。预期结果：不进行响应。

作为用户，可以发布文章，以便分享课程知识和技能沉淀 1.用户输入符合要求的文章。预期结果：提示发布成功。2.用户未输入文章。预期结果：提示文章内容不能为空。

作为用户，可以浏览文章，以便获取知识 1.用户点击正常文章列表的标题。预期结果：跳转到该文章的展示界面。2.用户点击已被发布用户删除的文章标题。预期结果：提示该文章已被发布者删除。

作为用户，可以加入课程，以便获取相关课程的学习内容 1.已登录用户加入课程后点击课程章节。预期结果：跳转至该课程章节的视频播放页面。2.未登录用户点击加入课程或课程章节。预期结果：跳转到登录界面。3.用户未加入课程就点击课程章节。预期结果：提示请先加入该课程。

作为用户，可以观看课程视频，以便学习课程内容 1.用户观看正常的视频。预期结果：视频正常播放。2.用户观看已被删除或视频文件缺失的视频。预期结果：提示视频不见了。

作为用户，可以浏览各项课程，以便获知可选的课程概览信息 1.用户进入课程中心浏览课程。预期结果：显示课程列表

作为用户，可以按照类别筛选课程，以便浏览特定类别的课程 1.用户点击某个课程分类。预期结果：显示该领域下的所有课程信息。

作为用户，可以选中并查看课程的大致内容，以便获知此课程的大概结构1.用户点击某个课程。预期结果：跳转至该课程的课程详情页面。

作为用户，可以评论课程，以便发表对此课程的看法 1.已登录用户输入符合要求的评论。预期结果：提示评论成功。2.未登录用户点击发布。预期结果：跳转到登录界面。3.用户通过评论进行sql、js注入攻击。预期结果：提示评论失败且无法造成影响。4.用户输入内容为空。预期结果：提示评论内容不能为空。

作为用户，可以根据关键字查询课程，以便查找想要了解的课程 1.用户输入正确的关键字后搜索。预期结果：显示包含该关键字的课程。2.用户输入关键字为空。预期结果：显示搜索内容不能为空。

作为用户，可以按照分类筛选文章，以便浏览特定类别的文章 1.用户点击某个文章分类。预期结果：显示该领域下的所有文章信息。

作为用户，可以发布文章，以便分享课程知识和技能沉淀 1.用户输入符合要求的文章。预期结果：提示发布成功。2.用户输入的文章未填写标题。预期结果：提示标题不能为空。3.用户输入的文章未选择分类。预期结果：提示未选择分类。4.用户输入的文章内容为空。预期结果：提示文章内容不能为空。5.用户输入的文章未填写简介。预期结果：文章简介不能为空。

作为用户，可以发表文章评论，以便与其他用户分享对文章的看法 1.已登录用户输入符合要求的评论。预期结果：提示评论成功。2.未登录用户点击发布。预期结果：跳转到登录界面。3.用户通过评论进行sql、js注入攻击。预期结果：提示评论失败且无法造成影响。4.用户输入内容为空。预期结果：提示评论内容不能为空。

作为用户，可以删除文章，以便删去错误或过时的文章 1.用户点击删除按钮来进行删除文章等相关操作。预期结果：提示删除成功。

作为用户，可以回复文章下的评论，以便对评论发表自己的看法 1.用户可以对文章下的相关评论进行自己的评论。预期结果：在该评论下出现自己的评论。2.用户未登录就想进行评论。预期结果：提示该用户请先登录。3.用户通过评论进行sql、js注入攻击。预期结果：提示评论失败且无法造成影响。4.用户输入内容为空。预期结果：提示评论内容不能为空。

作为用户，可以对文章点赞，以便表达对此文章观点的支持 1.用户可以对自己喜欢或者同意的文章进行点赞。预期结果：该文章的点赞数+1。2.用户未登录就想进行点赞。预期结果：提示该用户请先登录。2.用户进行奇数次点击点赞按钮。预期结果：该博文的点赞数+1。3.用户进行偶数次点击。预期结果：该博文的点赞数不变。

作为用户，可以提出问题，以便征集其他用户对问题的看法 1.用户可以提出问题发布在该网站上。预期结果：提示发布成功。2.用户未登录就想发布问题。预期结果：提示该用户请先登录。3.用户输入的问题未填写标题。预期结果：提示标题不能为空。4.用户输入的问题未选择分类。预期结果：提示未选择分类。5.用户输入的问题内容为空。预期结果：提示内容不能为空。

作为用户，可以删除自己发布的问题，以便删去错误发布的问题 1.用户可以把自己提出的问题删除。预期结果：提示删除成功。2.用户删除的问题已被管理员删除。预期结果：提示该问题已被管理员删除。

作为用户，可以回答其他用户发布的问题，以便发表自己对问题的看法 1.用户可以回答其他用户发布的问题。预期结果：提示回答成功，在该问题下方出现自己的答案。2.用户未登录就想回答问题。预期结果：提示该用户请先登录。3.用户通过评论进行sql、js注入攻击。预期结果：提示评论失败且无法造成影响。4.用户输入内容为空。预期结果：提示评论内容不能为空。

作为用户，可以删除自己发布的回答，以便删去错误发布的回答 1.用户可以删除自己发布的回答。预期结果：提示删除成功。

作为用户，可以查看自己的个人资料，以便了解自己个人资料的填写情况1.用户可以点击相关按钮来查看自己的个人资料。预期结果：弹出个人资料界面。

作为用户，可以编辑自己的个人资料，以便完善相关个人信息 1.用户可以点击相关按钮来编辑自己的个人资料。预期结果：弹出自己的个人资料界面。2.用户保存修改时，存在空的个人信息。预期结果：提示该个人信息不能为空。

作为用户，可以关注用户，以便及时获取相关用户动态 1.已登录用户点击关注按钮对其他用户进行关注。预期结果：提示关注成功，关注按钮变为取消关注按钮，用户的关注列表出现该用户。2.未登录用户点击关注。预期结果：跳转到登录界面。

作为用户，可以浏览问答列表，以便找到自己感兴趣的问题 1.已登录用户点击问题查看问题。预期结果：跳转到问题详细页面。2.未登录用户点击问题。预期结果：跳转到登录界面。

作为用户，可以根据分类筛选问答，以便浏览某一领域的问答 1.用户选择不同的分类标签。预期结果：显示该分类下的相关问题。

作为用户，可以根据关键词查找问答，以便精确地找到自己的目标。 1.用户输入正确的关键词。预期结果：显示包含该关键字的所有问题。2.用户输入错误的关键词。预期结果：显示未找到相关问题。3.用户进行sql或js注入攻击。预期结果：提示输入错误或无提示且无法造成影响。4.用户未输入关键词就点击搜索。预期结果：提示关键词不能为空或无反应。

作为用户，可以查看问答详情，以便详细了解问题 1.用户点击某个问题。预期结果：跳转至该问题的问题详情页面。

作为用户，可以参与课程考试，以便检验自己的学习成果。 1.已登录用户点击考试。预期结果：进入考试相关页面。2.未登录用户点击考试。预期结果：跳转到登录界面。

作为用户，可以给回答点赞，以便表示对回答内容的赞同 1.已登录用户对回答进行点赞。预期结果：该回答的点赞数+1。2.未登录用户对回答进行点赞。预期结果：跳转到登录界面。3.用户进行奇数次点击点赞按钮。预期结果：该回答的点赞数+1。4.用户进行偶数次点击。预期结果：该回答的点赞数不变。

作为用户，可以回复回答，以便发表自己对回答的看法 1.用户可以对问题下的回答进行自己的评论。预期结果：在该回答下出现自己的评论。2.用户未登录就想进行评论。预期结果：提示该用户请先登录。3.用户通过评论进行sql、js注入攻击。预期结果：提示评论失败且无法造成影响。4.用户输入内容为空。预期结果：提示评论内容不能为空。

作为用户，可以管理自己的回答，以便对回答进行修改、删除等操作 1.用户在个人页面查看已发布的问题回答。预期结果：显示该用户已发布的回答列表。2.用户点击修改按钮来修改回答。预期结果：显示修改页面。3.用户点击删除按钮来删除回答。预期结果：显示删除成功。4.用户未进行任何回答且进行查看。预期结果：提示还没有回答

作为用户，可以查看自己的上课记录，以便继续自己的课程学习 1.用户点击某个学习过的课程。预期结果：跳转到该课程上次学习时的视频播放界面。2.用户点击某个未开始学习的课程。预期结果：跳转到该课程的课程详情页面。

作为用户，可以管理自己关注的问题，以便进行查看、取关等操作。 1.用户点开自己的问题关注列表以查看自己已关注的问题。预期结果：弹出该用户所关注的问题列表。2.用户未关注问题就进行查看。预期结果：提示还没有关注问题。

作为用户，可以查看自己的关注对象，以便了解自己的关注情况 1.用户可以点击关注列表按钮以打开自己的关注列表查看自己关注的人。预期结果：跳转到关注列表界面。

作为用户，可以管理自己关注的用户，以便对关注用户进行查看、取关等操作 1.用户点开自己的用户关注列表以查看已关注的用户。预期结果：显示该用户的关注用户列表。2.用户未关注其他用户就进行查看。预期结果：提示还没有关注的用户。

作为用户，可以浏览消息列表，以便查看最新的消息动态 1.用户可以点开消息界面以查看相关消息的最新动态。预期结果：跳转到消息界面。2.未登录的用户想打开消息列表。预期结果：提示该用户未登录，并且无法显示已关注的相关动态。

作为用户，可以关注问题，以便获取该问题最新的回答 1.用户对问题进行关注。预期结果：提示关注成功，关注按钮变为取消关注按钮，用户的关注列表出现该问题。2.用户未登录就想进行关注。预期结果：提示该用户请先登录。

**1.3 原型图绘制**

课程中心





共享中心（文章博客部分）





问答天地

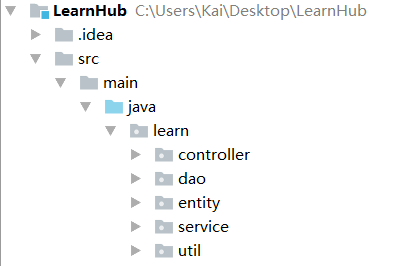


用户空间



**2 编码实现阶段**

**2.1 项目结构**



后端架构采用Spring + Spring MVC + MyBatis框架，按照MVC分层设计：

* 模型（Model）代表数据控制器。数据的读取，插入，更新都是由模型来负责。
* 视图（View）是展示给用户的最终页面。视图负责将数据以用户友好的形式展现出来。
* 控制器（Controller）是模型，视图以及其他任何处理 HTTP 请求所必须的资源之前的中介

controller做一些service前后的数据处理，利用service完成更高的操作。作为View和Model中间层。

dao用于操作数据库，针对表进行增删改查最基础的操作。

entity为实体类，作为MVC的M是为view服务的，传递和接收View的数据。model与数据库表的字段映射，并且也添加前端需要的字段。。

service利用dao进行业务上的处理，主要是展现需要开放出去的接口，一个service应当只调用自己对应的dao，如果需要调用其他实体的操作，需要将其他实体的service引入来使用。

util为工具包。

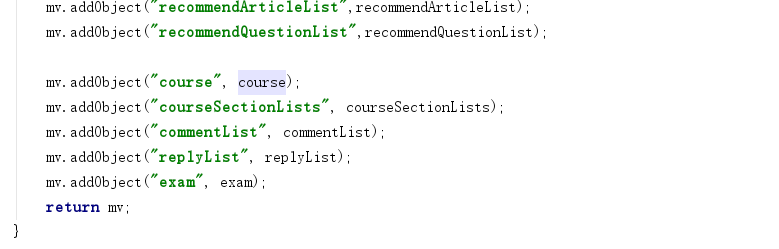
**2.2 实现方案**

* 当用户点击课程主页，Controller控制器拦截请求，调用Service层从数据库读取课程数据，传到前端。



* 用户点击具体课程，Controller控制器拦截请求，调用Service层从数据库读取数据，判断用户是否学习过该课，读取学习信息，同时将课程详细数据与相关类型的课程与回答数据传到前端。





* 用户点击文章页面，负责article的Controller控制器拦截请求，调用Service层从数据库读取文章数据，传到前端。当Controller接受到前端传来的发布新文章、评论或回复等POST请求，分别调用相应方法。



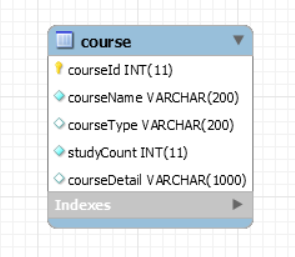


* 用户点击问答模块，则由负责question的Controller控制器拦截请求，调用Service层从数据库读取提问与回答数据，传到前端。对用户的编写提问、回答或是点赞等请求，调用控制器对应方法。



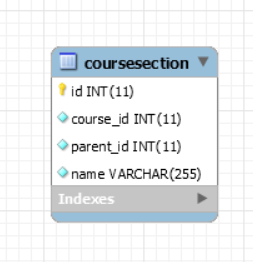
**2.3 数据库ER图设计**

* 课程表



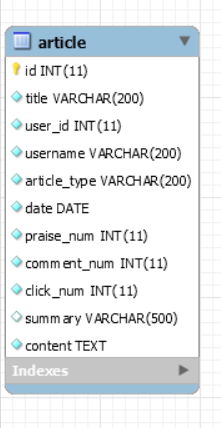
包含id，课程名，课程类型，学习人数，课程详情字段。

* 课程章节表



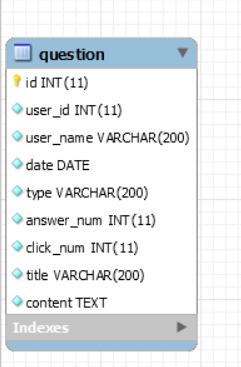
包含id，课程id，父章节id，章节名字段。

* 文章表



包含id，标题，发表用户id，用户名，文章类型，日期，点赞数，评论数，点击数，摘要与内容字段。

* 问题表



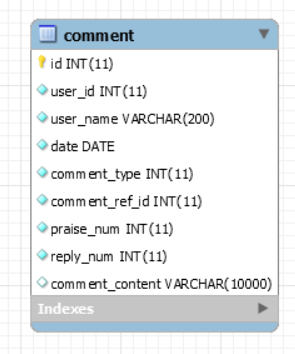
包含i发表用户id，用户名，问题类型，日期，回答数，点击数，标题，内容字段。

* 回答表



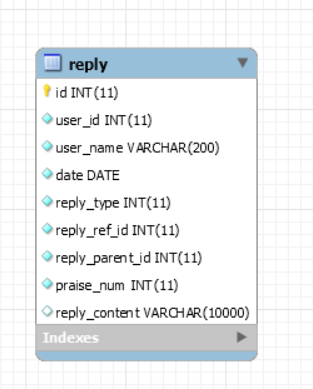
包含id，发表用户id，用户名，对应问题id，日期，点赞数，评论数，内容字段。

* 评论表



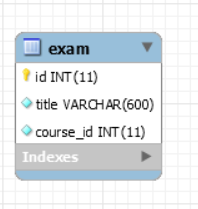
包含id，发表用户id，用户名，日期，评论类型（文章评论或回答评论），评论引用id，点赞数，回复数，内容字段。

* 回复表



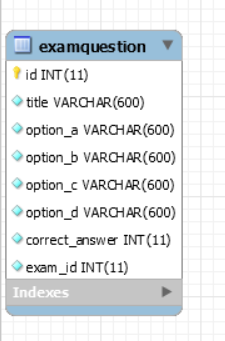
包含id，发表用户id，用户名，日期，回复类型，回复引用内容id，父id，点赞数，内容字段。

* 考试表



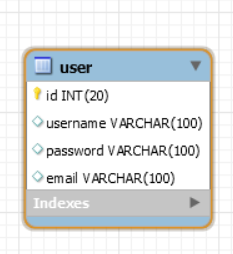
包含id，考试标题，对应课程id字段。

* 考试题目表



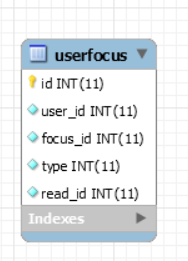
包含id，考试题目，选项内容，正确答案，对应考试id字段。

* 用户表



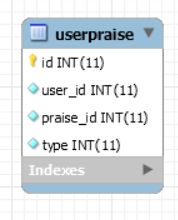
包含id，用户名，密码，邮箱字段。

* 用户关注表



包含id，本用户id，关注id，关注类型（用户或提问），阅读情况字段。

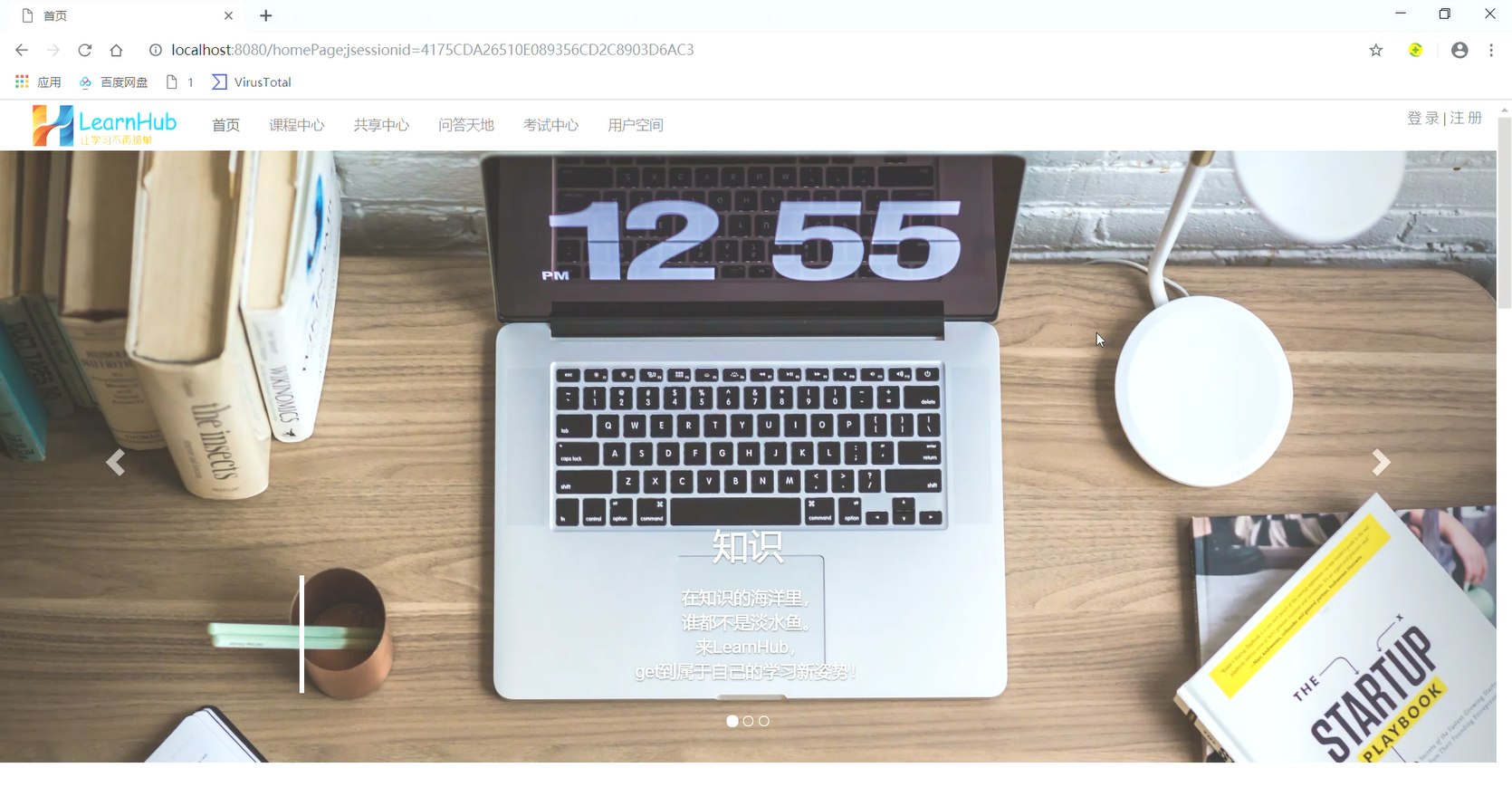
* 用户点赞表



包含id，用户id，点赞内容id，点赞类型（对文章、回答还是评论点赞）字段。

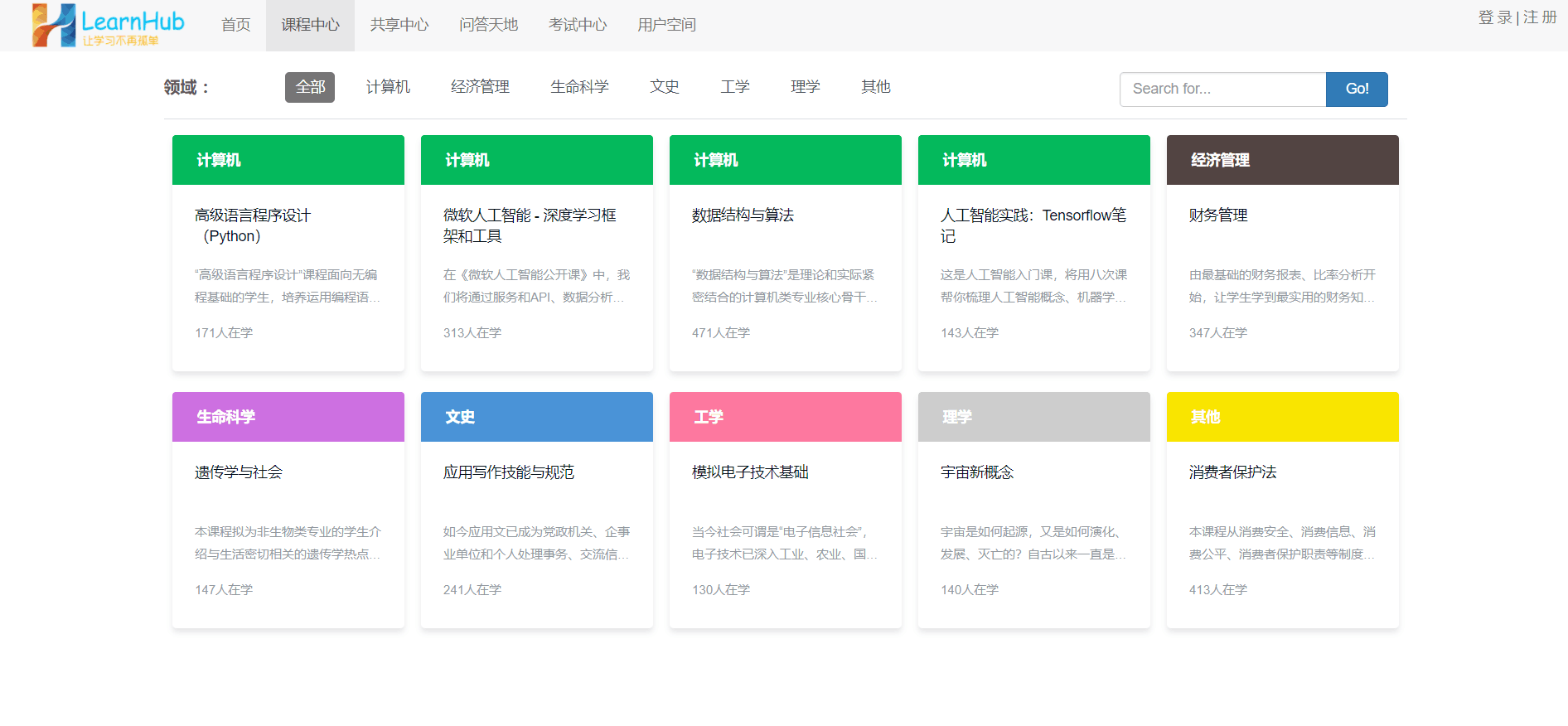
**3 项目成品展示**

首页：包含推荐课程内容、精选文章、问答等





课程中心模块

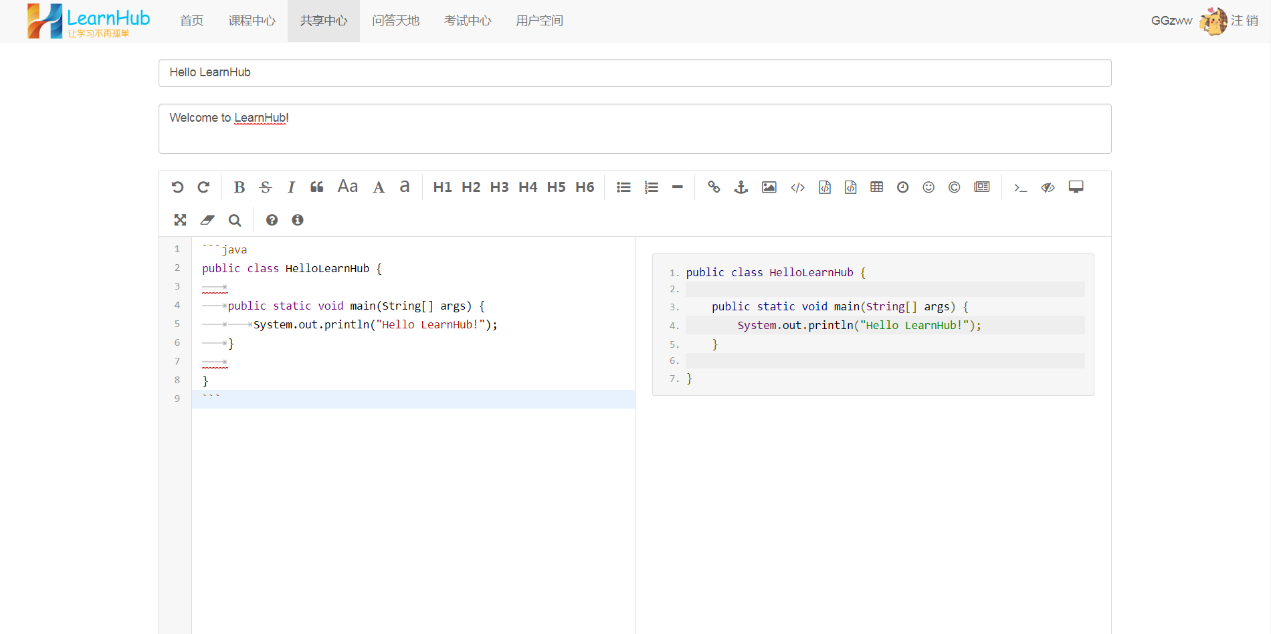




共享中心模块



共享中心中，编辑博客页面



问答天地模块



用户空间模块

