

南京信息工程大学

本科生毕业论文(设计)



题 目 基于驱动的技术学习平台设计与实现

学生姓名 夏佳

学 号 2 0121344035

学 院 计算机与软件学院

专 业 软件工程

指导教师 王定成

二〇一 六 年 五 月 二十 日

声 明

本人郑重声明：

- 1、 持以“求实、创新”的科学精神从事研究工作。
- 2、 本论文是我个人在导师指导下进行的研究工作和取得的
研究成果。
- 3、 本论文中除引文外，所有实验、数据和有关材料均是
真实的。
- 4、 本论文中除引文和致谢的内容外，没有抄袭其他人或
其他机构已经发表或撰写过的研究成果。
- 5、 其他同志对本研究所做的贡献均已在论文中作了声明
并表示了谢意。

作者签名：_____

日 期：_____

目 录

1. 绪论	1
1.1 选题的背景与意义	1
1.1.1 选题背景	1
1.1.2 选题意义	2
1.2 相关技术简介	2
1.2.1 HTML	2
1.2.2 CSS	3
1.2.3 Sass	3
1.2.4 Compass	3
1.2.5 JavaScript 语言	4
1.2.6 Materialize 框架	4
1.2.7 LeanCloud 框架	4
1.2.8 NodeJS Express 框架	5
2. 软件分析与设计	5
2.1 系统分析	5
2.1.1 可行性分析	6
2.1.2 需求分析	6
2.2 软件功能模块设计	7
3. 数据库设计	9
3.1 数据库 E-R 关系	9
3.2 数据库表设计	12
4. 系统详细实现	15
4.1 后端实现	16
4.1.1 数据服务连接	16
4.1.2 数据库创建	17
4.1.3 数据库操作	17
4.1.4 网络请求	17
4.2 用户模块	17
4.2.1 注册	17
4.2.2 登录	18
4.2.3 找回密码	19

4.2.4 修改密码	20
4.2.5 退出登录	20
4.3 课程模块	21
4.3.1 课程展示	21
4.3.2 课程学习	22
4.3.3 课程进度	23
4.4 社区模块	24
4.4.1 发布状态	24
4.4.2 状态列表	25
4.4.3 状态评论	25
4.4.4 朋友圈	26
4.5 任务模块	27
4.5.1 任务列表	27
4.5.2 任务提醒	28
5. 系统测试.....	29
结束语	30
参考文献.....	31
致谢	31

基于驱动的技术学习平台设计与实现

夏佳

南京信息工程大学计算机与软件学院，江苏 南京 210044

摘要：随着 MOOC 理念在全球的推广与普及，各类技术学习网站和社区兴起，他们以开放免费的网络课程模式吸引了许许多多的求知者。但是现如今，各大网站的学习资源冗杂，很多初学者常常半途而废。本网站旨在给用户提供在线学习课程与良好的学习交流社区，使其学习更轻松。网站基于 B/S 架构，使用 Materialize 前端框架与 LeanCloud 的 Baas 框架开发，使用 JavaScript 进行前后端通信。用户使用该网站可以查看、学习相应的课程，并且可以在课程中心查看自己的学习进度。用户还能在社区提问或者回答他人的问题。同时，网站为用户制定学习任务，使用邮件推送的方式督促用户学习。经测试，该网站用户体系完善，界面美观，交互友好，方便学习交流。

关键词：前端；在线学习；B/S；驱动式

Design and implementation of driven technology learning platform

Jia Xia

School of Computer and Software, NUIST, Nanjing 210044, China

Abstract : With the popularization of MOOC in the world , various technical and community learning sites rise , they are in a new model curriculum attracts many knowledge seekers , and that is for the public free open online courses mode. However , resources miscellaneous, many beginners often fall by the wayside . This site aims to provide users with e-learning courses and a good learning communities. Website is based on B / S structure , use Materialize and Leancloud framework , the front and rear ends communicate with JavaScript. Users can use the site to view, study courses , and you can view your progress in the course center. Users can ask questions and communicate together in the community. Meanwhile, the website develops learning tasks for users, and send email to remind them. After testing , the site user system and interface is beautiful , interactive and friendly, easy to learn and communicate.

Key words : Frontend; Online Learning; B/S; Driven

1. 绪论

1.1 选题的背景与意义

1.1.1 选题背景

2012 年起，免费课程开始陆续地出现在美国各个顶尖大学设立的线学习平台上。自此，MOOC 的理念以不可抵挡之势，席卷了全球。2012 年被称为“MOOC 年”。然而，MOOC 的发展历史非常的短暂，从它的孕育到如今大范围的影响，仅仅经历了 50 多年的时间，它意味着计算机开始慢慢与其他领域融合在一起。MOOC 一时间近乎疯狂的崛起，让人们体会到网络已经慢慢渗透到了教育之中，其全新的教学模式冲击了传统的高等教育，甚至会影响改变传统教育的格局。

2013 年初，中国多所大学加入到 MOOC 建设实践中，加盟 edX、Coursera，建立了自己的在线教育平台。国内有的学者认为 MOOC 在中国的发展可能会改变高等教育的格局，但也有人认为，它对中国的教育并不会产生巨大的冲击。因为那只是网络教育发展至今衍生出来的一种新形式而已。要知道，我国的网络教育早在 1994 年的时候就开始了，经过了二十多年的发展，已经有了相当大的规模。

我国最早开始关注研究 MOOC 的学者是南京师范大学的焦健利教授，他在《说慕课的到来去脉》时，第一次把 Massive Open Online Course 翻译成了“慕课”。从字面上来理解，MOOC 就有大规模的、开放的、网络课程这几个特点。MOOC 是一种超大的巨型课程，全新的知识传播模式和学习方式相比传统的课堂教育有许多优点。首先，这些在线平台打破了传统课堂时间和地域的限制，每个人都可以自主开展学习。其次，用户的学习时间可以自由分配，利用空闲时间学习新的技术知识。最重要的是，通过在线学习平台，用户可以和来自世界各地的小伙伴一起分享和交流，产生思想的碰撞，擦出智慧的火花。

北京青年报曾报道，在 2013 年 10 月 17 日，edX 平台和学堂在线开发了《电路原理（一）》课程，当天选择这门课程的用户达到了 10088 名，并且他们来自于一百多个国家与地区，这样的数据已经超过了过去 20 年清华大学校内选修这门课程的学生总数了。另外有互联网数据显示，从 2012 年开始，一年多的时间里，网易公开课上的在线课程数量迅猛增加，达到了 12000 多门，是之前的一倍。同时，每天大约有 460 万用户通过各种方式访问学习这些课程，其中，大约有 100 万的用户是学生群体。根据这些数据不难看出，MOOC 近年来在国内的取得了巨大的发展。

1.1.2 选题意义

MOOC 在中国发展已经是第 4 个年头，其影响不容小觑。目前，中国的大学教育面临着改革与创新。不仅仅是教育技术的革新，更加是教育观念、教育方式等方面的变化。首先，传业授道解惑，不再局限于老师课堂上的谆谆教诲，老师可以在网上授课，可以即时发布学习资源。从前，一个老师一次只能给几百人又或几十人授课，现在，学习资源变成了“活的”，成千上万的人可以一起学习同一门课程，学生能够足不出户聆听国内外顶尖大学的讲座。其次，MOOC 理念重新定义了老师与学生的概念，在这样开放的平台上，所有人既可以是老师又可以是学生，它所提供的社区、论坛，让他们互相帮助互相学习。MOOC 把免费的、最好的课程和教育带给地球最偏远的角落的人们，帮助他们的职业生涯，并且帮助他们扩充知识和拓宽人际网络。对于一些没有机会没有条件接受高等教育的人，这样的平台对于他们无疑是雪中送炭，并且从一定程度上来讲，MOOC 的发展也促进了教育公平。

在全球 MOOC 的浪潮之下，我国也致力于推出中国本土的 MOOC。正如 Coursera、Udacity、edX 等课程网站的兴起一般，国内也兴起了一大批提供开放网络课程的平台。其中像 MOOC 学院、学堂在线、网易云课堂等等是大家耳熟能详的。它们都提供着丰富多彩的课程，然而对于基础薄弱的初学者来说，丰富多彩变成了纷繁复杂，他们无法准确地定位哪些才是他们真正需要的课程，因此而白白浪费时间。另外，多数大型的学习平台都没有将响应式适配到手机端，因此从手机端浏览器打开的无法有良好的体验进行学习。

为了弥补现今在线学习平台的不足，搭建了一个驱动式技术学习平台是有意义的。本网站想要提供适合 IT 初学者的入门课程，设置学习任务，帮助他们轻松学习，轻松入门。首先，整合优秀的、合适的学习资源，避免用户在选择课程时的不知所措。其次，网站提供一个开放的社区，用户可以浏览话题，提问以及解决他人的问题，这样，就不会由于没有交流渠道而产生一个个“信息孤岛”，甚至导致用户放弃继续学习一门技术。初学者不断地学习到知识，认识学习过程中的不足，形成良性的循环，更快更轻松的入门，最终修炼成业界大牛就是本网站建立的初衷。

1.2 相关技术简介

1.2.1 HTML

HTML (HyperText Markup Language)，顾名思义，是描述 Web 文档的一种标记语言，是所有网站内容的核心语言。浏览器读取 HTML 文件，将其渲染成可视化的网页。通过结合其他 Web 技术，可以构建出令人赏心悦目、功能强大的网页。不同的 HTML 元素组合在一起，形成一个网页的基本骨架。元素属性一般以“名称：值”的形式成对出现。HTML5 是 HTML 定义标准的最新发展。

1.2.2 CSS

CSS (Cascading Style Sheets) 被称为层叠样式表，是网页的表现层。如果说 HTML 是网页的骨架，那 CSS 就是网页的装潢。它最主要的目的就是使文档的内容与显示分隔开来，增加了文件的可读性，使文档结构更加灵活等。用户可以根据自己的需求来调整网页上元素的背景，大小，颜色，并且利用 CSS 实现对网页中对象的位置排版控制，精确到像素级别。使用 CSS 可以让代码更加简洁，提高网页访问速度，有利于突出重点，适合搜索引擎抓取等。

CSS3 是 CSS 定义标准的最新发展。

CSS 规则如：选择器 {属性 1: value1; 属性 2: value2; ... 属性 N: valueN }

1.2.3 Sass

CSS 不能进行编译，对于程序员来说，没有变量，没有条件语句，只是单纯的表述，写起来 CSS 来就很费力。后来就有人把编程元素融入其中，于是就有了 CSS 预处理器，Sass 就是其中的一种。它基于 Ruby，为开发者提供了很多便利的写法，比如在 CSS 的基础上添加了扩展功能，包含变量、嵌套、混合等，从而让代码变得简单可维护。在编写比较大的网页时，你再也不用担心类名重复，不用浪费时间在样式表结构上，不用一段代码重复好几遍编写。Sass 还提供了一些内置的颜色函数，开发者可以加以利用，制作想要的颜色。同时 Sass 的高级用法支持条件判断自定义函数等，这让前端开发在不使用 JavaScript 的情况下就能够去处理逻辑判断，无疑是为开发者带来的福音。

1.2.4 Compass

Compass 是一个开源的 CSS 框架，Compass 封装了一系列模板来补充 Sass 的功能，它和 Sass 的关系就类似于 JS 与 JQuery 的关系。Compass 共有 5 个内置模块 (reset、css3、layout、typography、utilities)，分别提供不同的功能。以 utilities 为例，它提供某些不属于其他模块的功能，比如清楚浮动、表格、制作雪碧图等等。Compass 的安装也非常简单，只需要先安装 Ruby 环境，然后在命令行中输入安装命令，正常情况下，Compass 连同 Sass 就安装好了。在开发过程中，手动去编译 Sass 文件非常繁琐，你可以进入该项目目录相应文件夹，执行 compass watch 命令，它会帮助我们监视每一个 Sass 文件，一旦产生变化，就会自动执行编译命令。当然，你也可以使用 compass compile -force 去编译所有的 Sass 文件。一般来说，编译出来的样式表会带有很多注释信息，如果你不需要可以对其进行压缩，只要加上 --output-style 参数即可。Compass 还有许多神奇的用法，此处不再赘述。

1.2.5 JavaScript 语言

JavaScript（“JS”的简称）是一个完整的动态编程语言，当应用到 HTML 文档时，可以在网站上提供动态交互。它是由 Mozilla 项目的共同创始人布兰登·艾克创建的。

JavaScript 几乎可以应用到任何开发环境中，适用范围非常广。小到一些很简单的功能，如旋转木马，图像画廊，波动的布局，和响应按钮点击等等。在你使用经验丰富之后，你用它来创建游戏，动画 2D 和 3D 图形，开发完全成熟的数据库驱动的应用程序，甚至更多神奇的东西！

作为脚本语言，其本身不需要编译，能直接在浏览器中运行。这样的特点会造成他的一些局限性，但也挡不住它强大的功能带来的便利。JavaScript 可以实现在客户端进行数据验证的功能，免去了用户端浏览器通过网络传输到服务器上进行提交验证的工作，大大减少的网络传输。并且，JavaScript 让用户能够灵活得控制 DOM 对象，提供友好的用户体验。

1.2.6 Materialize 框架

Materialize 是一个基于谷歌的全新设计语言 Material Design 而开发的一个现代响应式的前端框架，是由四个卡内基梅隆大学学生开发的。界面秉承着更干净、更简约的理念。Material Design 是由谷歌创建和设计的，是成功结合了设计经典原则与创新技术的设计语言。谷歌的目标是开发设计一个能够在任何平台的拥有统一的用户体验的系统。

Materialize 定义了空间和运动的关系。该技术的灵感来自纸和墨水，这样的灵感被利用以促进创造力和创新。表面和边缘提供了熟悉的视觉线索，让用户快速了解超越物理世界的技术。大胆的、形象的元素和组件，如网格，字体，颜色和图像，不仅在视觉上令人愉快，而且显得有层次感，有意思。突出不同的动作和组件为用户创建一个可视化指南。

1.2.7 LeanCloud 框架

LeanCloud 是国内针对移动应用的一站式云端服务，包括数据存储、实时消息和推送、统计分析，并且全面支持 iOS、Android、web 等多个平台。让开发者抛弃了后端开发的负担，有良好的用户体验。

LeanCloud 的数据存储对于 BaaS 而言，不用再理会「应用服务器」、「数据服务器」，不再担心流量和压力。对于数据库的存储只要满足「JSONObject」这一基本格式。采用 SQL 语句分析应用数据，文件存储采用国内全网加速 CDN，支持多种类型文件。并且提供全平台的 SDK 支持，包括 iOS、Android、Windows Phone、Unity 3D、Cocos2D-x 以及 JavaScript。而它的云引擎服务，能让你把代码部署到 LeanCloud 云端来执行，灵活

应对各种需求。并且支持定时任务，beforeSave 等 Hook 调用，支持在线编辑，本地部署，方便调试。提供成熟的命令行工具，高效的部署、发布、回滚引擎。

1.2.8 NodeJS Express 框架

NodeJS 是 JavaScript 语言向后端的扩展，突破了 JavaScript 只能运行在浏览器沙盒的一大限制，NodeJS 开发团队通过 ChromeV8 解释引擎将这门优秀的语言延伸至后端，解决了 JavaScript 非编译而是解释性语言特性所带来的副作用。

Express 是一个使用 NodeJS 开发的 Web 服务器应用程序，它为网络和移动应用提供了许多强大的功能。由于 NodeJS 有非常丰富的第三方包，Express 也可以应用到这些包，所以拓展性非常强。Express 具有非常丰富的 HTTP API，并且 API 也十分友好，是构建强大的 Web 后端服务器的不二之选。从性能上看，Express 继承了 NodeJS 非多线程异步的特点，避免了多线程环境切换的性能开销，采用异步事件的方式，概念更抽象却更加适合开发者理解。

2. 软件分析与设计

2.1 系统分析

本网站基于采用 MVC 的设计模式，由浏览器端和后端两部分组成。数据库应用 LeanCloud 的 BaaS 框架，后端应用基于 Express 的 NodeJS 框架，主要实现数据服务的连接，数据创建与查询，网络请求的处理、响应，用户对象的安全管理等。浏览器端应用 Materialize 的前端框架，建立了多个数据模型，包括用户、课程列表、我的课程状态、学习任务列表、社区状态列表、话题发布、评论、任务等。每个数据模型包含多个数据字段，用对象来表示数据，存储到相应的类中。

模型建立完成之后，需要创建其相对应的视图。主要视图包括：用户登录与注册视图、课程展示视图、我的课程视图、学习任务视图、社区发帖视图、帖子总览视图、评论视图、内容回复视图、学习任务视图等。数据模型与视图进行绑定，一旦数据的变化，就会反应到界面上来。

控制器的主要作用是对视图上用户的操作进行处理。例如对用户点击操作的处理等。本系统中建立若干控制器，例如注册登录控制器、列表控制器、详情页面控制器等等。这些控制器通过用户的点击事件触发，发送请求，这都需要为各个视图绑定监听事件才能完成。

系统预期实现目标如下：

1、用户注册登录网站，在网站首页查看所有课程，选择开始学习相应课程，随即跳转到课程的详细视图中。用户点击上一节、下一节按钮学习各章节，并且由网页记录下用户学习的进度，方便下次继续学习。用户也可以登录个人中心，查看学习结束的课程和正在学习的进度，并且查看学习任务完成的情况。

2、用户可以进入社区，查看社区中所有人提的问题，并且可以查看、发表评论。也可以发布自己的问题，等待他人的回答，回复他人的回答。

2.1.1 可行性分析

1. 经济可行性

本网站使用 WebStorm 免费 30 天试用期的开发工具，以及 Materialize 与 LeanCloud 开放性的框架，不需要产生任何费用。由于网站开发没有使用后端语言，不需要为服务器配置环境，因此对服务器的要求并不是很高。就说国内服务器提供商而言，大多提供了比较便宜的方案，长达半年到 2 年的免费试用时间，静态托管类型的大多数是免费，所以几乎不需要经济付出。

2. 技术可行性

本网站开发使用的主要技术有 HTML，CSS，JavaScript，Materialize 框架、LeanCloud 框架和 NodeJS 框架。这些技术都比较成熟，有很多可以参考的优质学习资源，而且难度不大，初学者容易上手。而 Materialize 框架和 LeanCloud 框架在其官网上都有详细的文档，只需有一点编程基础，就能很快的理解使用。而在大学期间多多少少都有涉猎了前端的相关知识，所以在技术上是可行的。

3. 结论

本网站在经济和技术上都是完全可行的，交互良好，界面美观，方便用户学习与交流。

2.1.2 需求分析

1. 功能需求

本网站功能分为五大模块，用户模块，课程模块，社区模块，课程进度模块，学习任务模块。

用户模块包含用户注册、登录、退出、修改密码以及丢失密码，通过第三方找回功能五大功能，用户登录之后才能进行学习与交流。

课程模块在首页以及课程页展示课程信息，用户在选择相应课程后会跳转到课程详情页面，进行学习。

课程进度模块展示用户的学习进度，包含已学完的课程信息与正在学习的课程信息，同时用户可以选择重新学习与继续学习。

社区模块提供用户浏览状态，发表状态，评论状态的功能。用户可以在微型社区和其他用户分享交流。

学习任务模块提供一些学习任务以及其完成情况，对于未完成的任务的用户，将推送提醒邮件到其邮箱，从而驱动用户去更好地学习。

2. 性能需求

由于本网站需要加载很多图片资源，因此需要降低加载的压力，对于图片首先就要进行压缩，为了减少 HTTP 请求的次数，可采用制作雪碧图的方式，提高网站的加载速度。

2.2 软件功能模块设计

首先，本网站给分为两个部分：浏览器端和后端。浏览器端负责展示页面与后端传来的数据，后端负责响应用户请求，并且定时扫描用户任务完成情况，提醒用户。

后端功能模块主要包括实现数据服务的连接，类的创建，对象的创建与查询，处理网络请求，返回数据，发送邮件，用户对象的安全管理。后端根据浏览器端的相关功能提供相应的数据给浏览器。

浏览器端主要包含四个模块：用户模块，课程模块，社区模块，任务模块。用户模块分为用户注册、用户登录、修改密码、通过第三方找回密码、退出登录。课程模块实现了课程展示，课程学习，课程进度展示的功能。社区模块提供了用户发布状态，状态评论，状态回复，关注他人查看朋友圈，查看关注详情的功能。任务模块，展示学习任务的完成情况，网站服务器会定时遍历用户的课程学习情况，并发送提醒邮件。用户在社区可以与其他用户交流分享学习心得，并且关注他人。驱动式技术学习平台浏览器端功能模块如图 2-1 所示。

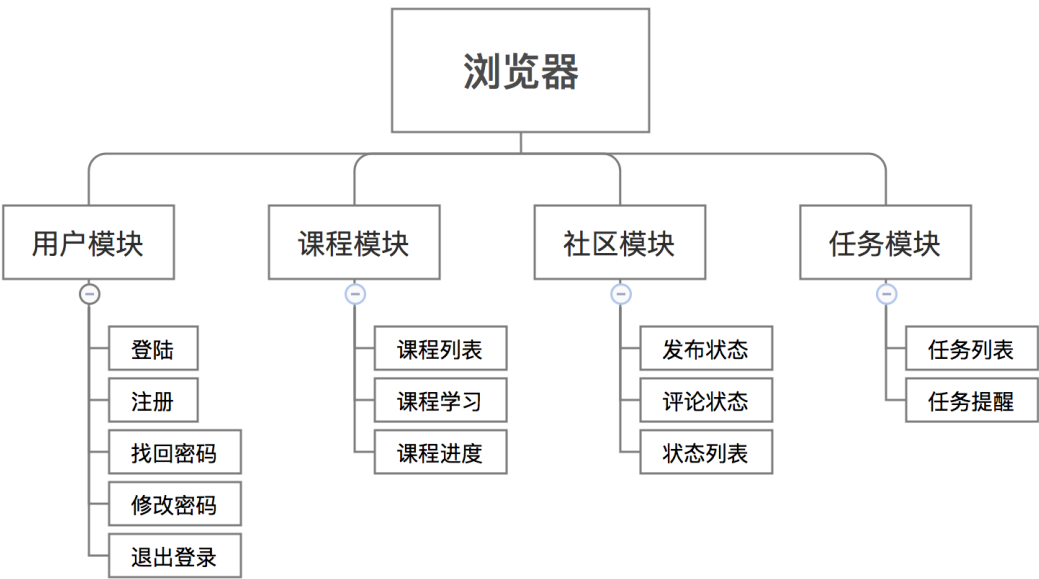


图 2-1 浏览器端功能模块图

服务器接口模块图如图 2-2 所示。

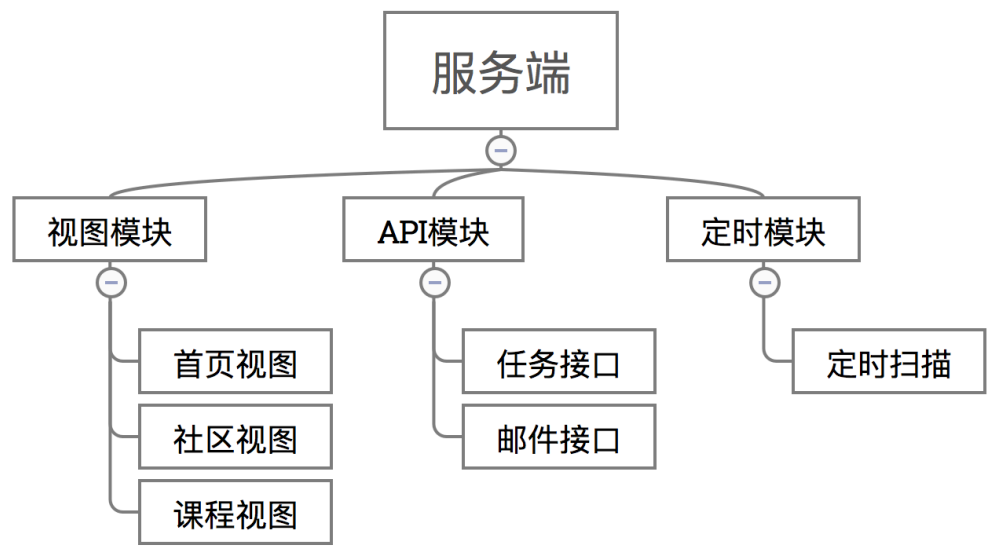


图 2-2 服务端接口模块图

网站操作流程如图 2-3 所示。

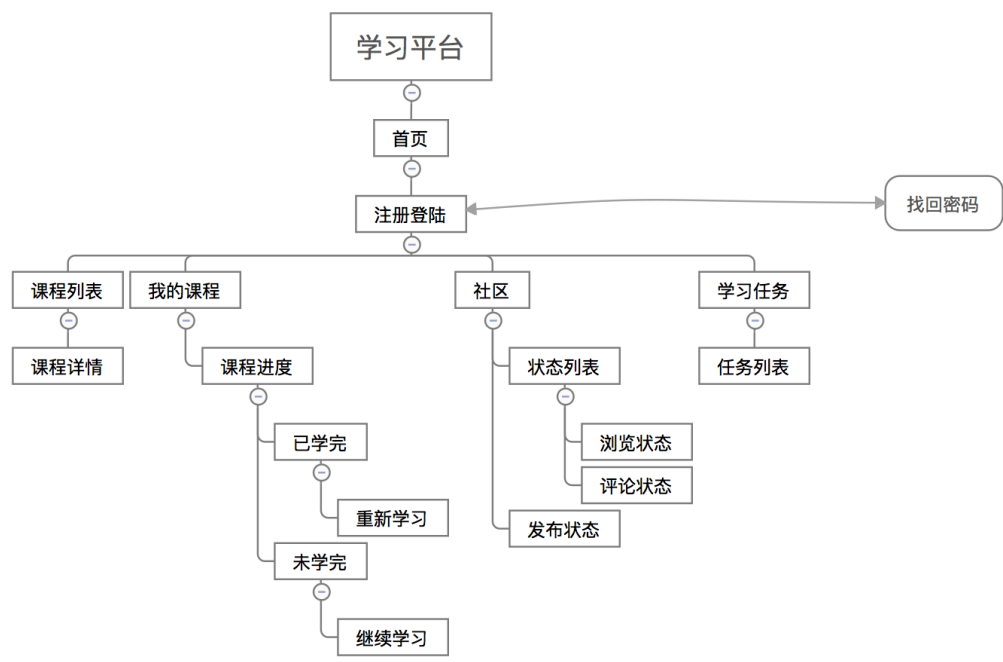


图 2-3 网站操作流程

3. 数据库设计

3.1 数据库 E-R 关系

在对网站进行分析过后，从数据的完整性、安全性以及合理性考虑，可以抽象出 7 个实体对象，分别是用户信息实体，课程实体，学习记录实体，状态实体，评论实体，任务实体，任务记录实体。

1. 用户信息实体（包含用户 Id、用户名、邮箱、密码、邮箱验证标识）如图 3-1 所示。

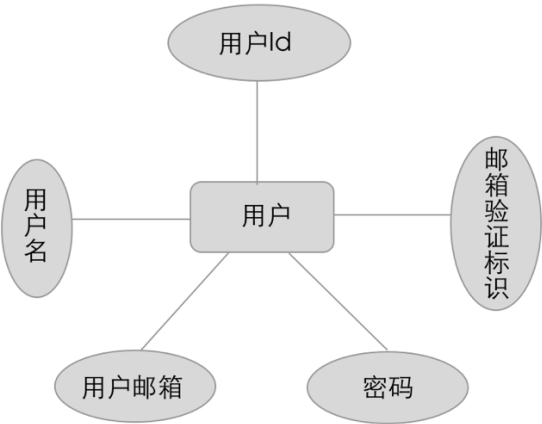


图 3-1 用户实体信息图

2. 课程实体（包含章节 Id、章节标识、标题、课程描述、课程章节内容、代码示例）如图 3-2 所示。

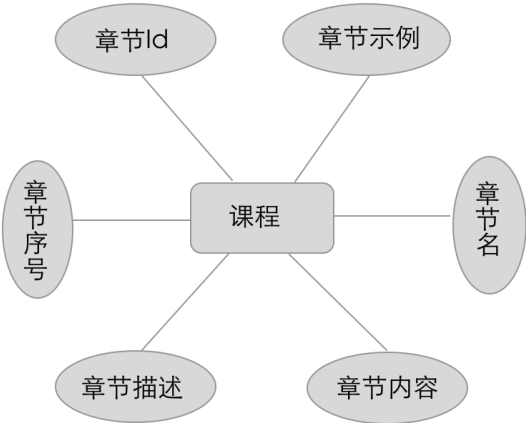


图 3-2 课程实体图

3. 学习记录实体（包含记录 Id、用户名、用户学习的课程名、学至的章节、是否学完标识）如图 3-3 所示。

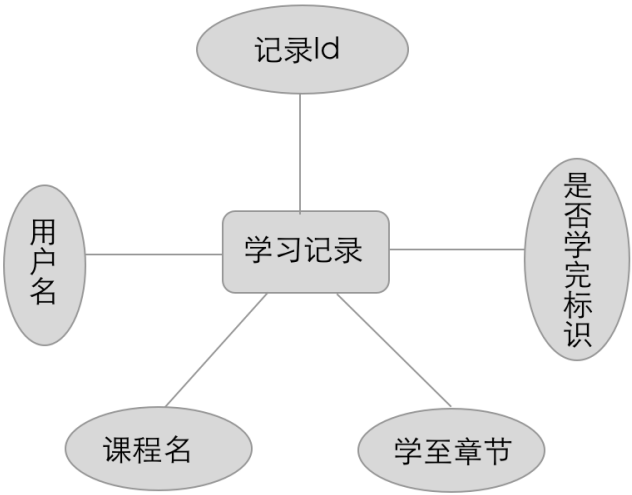


图 3-3 学习记录实体图

4. 发表的状态实体（包含状态 Id、状态内容、状态发布者的 Id）如图 3-4 所示。

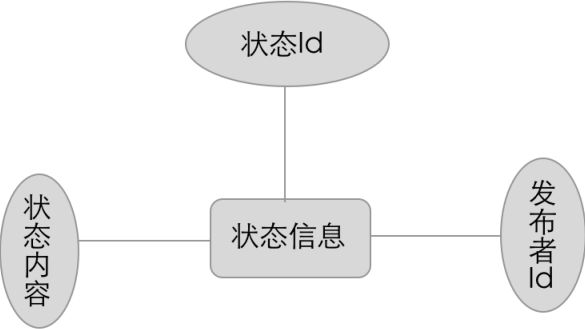


图 3-4 状态实体图

5. 用户提交的评论实体（包含评论 Id、评论者姓名、评论内容、该评论对应的状态），如图 3-5 所示。

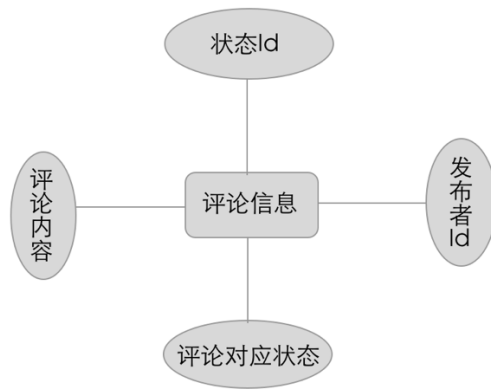


图 3-5 评论实体图

6. 任务实体（包含任务 Id、任务名称），如图 3-6 所示。



图 3-6 任务实体图

7. 任务记录实体（包含记录 Id、对应的任务 Id、对应用户 Id、任务名称、任务完成标识），如图 3-7 所示。

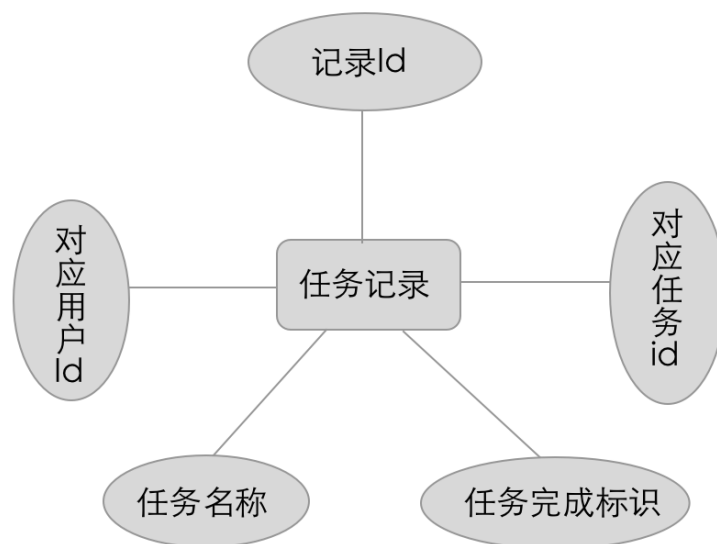


图 3-7 任务记录实体图

各实体之间关系图：

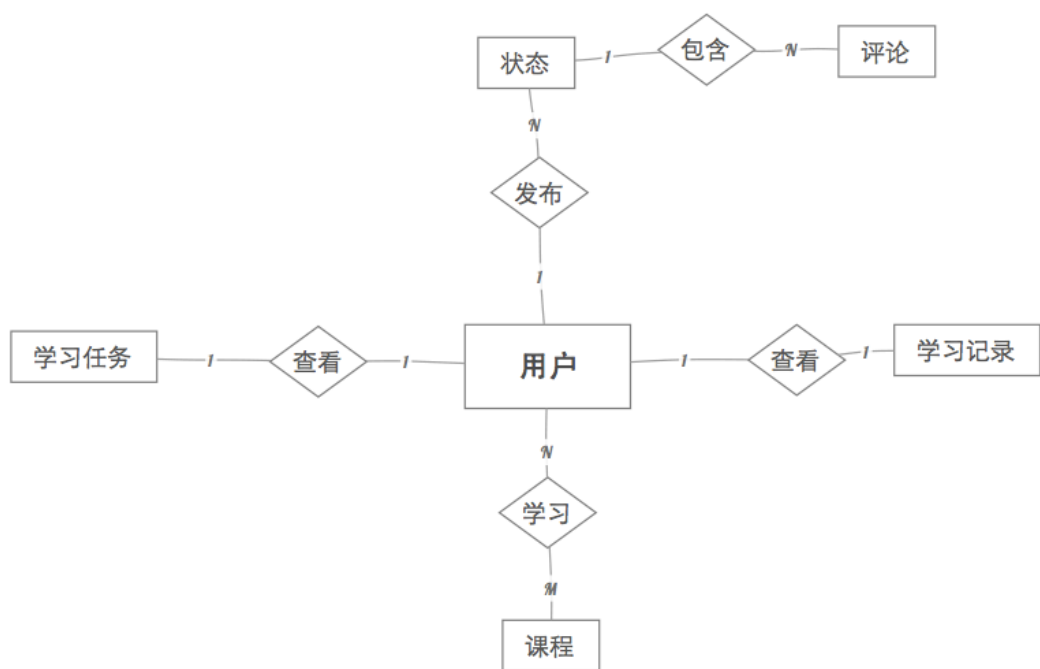


图 3-8 实体关系图

3.2 数据库表设计

E-R 图展示了本网站各个实体间的关系以及各实体包含的属性，接下来将根据此模型创建相应的数据库。本网站使用云端数据库 LeanCloud，其特点就是无需提前在工程中定义数据表与相关字段，除了课程资源，其余都可动态生成。以下就是各个表的具体信息、

1. 用户信息表（_User）组成字段有：用户 Id（objectId）、用户名（username）、邮箱（email）、密码（password）、验证标识（emailVerified）。如表 3-1 所示。

表 3-1 用户信息表

名称	数据类型	是否为主键	允许为空
objectId	string	是	否
username	string	否	否
email	string	否	否
password	boolean	否	否

2. 课程实体表（JS）（任取其中一门课程为例）包含章节 Id（objectId）、章节序号（order）、章节名（title）、章节描述（description）、章节内容（content）、章节示例（demo）。如表 3-2 所示。

表 3-2 课程实体表

名称	数据类型	是否为主键	允许为空
objectId	string	是	否
order	string	否	否
title	string	否	否
description	string	否	否
content	string	否	是
demo	string	否	是

3. 学习记录实体表 (Record) 包含记录 Id (objectId)、用户名 (user)、用户学习的课程名 (courseKind)、学至的章节 (chapter)、是否学完标识 (finish)。如表 3-3 所示。

表 3-3 学习记录实体表

名称	数据类型	是否为主键	允许为空
objectId	string	是	否
user	string	否	否
courseKind	string	否	否
chapter	string	否	否
finish	Boolean	否	是

4. 状态实体表 (_Status) 包含状态 Id (objectId)、状态内容 (message)、状态发布者的 Id (source)。如表 3-4 所示。

表 3-4 状态实体表

名称	数据类型	是否为主键	允许为空
objectId	string	是	否

续表 3-4:

名称	数据类型	是否为主键	允许为空
message	string	否	否
source	pointer	否	否

5. 评论实体表 (Comment) 包含评论 Id (objectId)、评论者姓名 (commenter)、评论内容 (content)、评论对应的状态 (status)。如表 3-5 所示。

表 3-5 评论实体表

名称	数据类型	是否为主键	允许为空
objectId	string	是	否
commenter	string	否	否
content	string	否	否
status	string	否	否

6. 任务实体表 (Task) 包含任务 Id (objectId)、任务名称 (TaskName)，如表 3-6 所示。

表 3-6 任务实体表

名称	数据类型	是否为主键	允许为空
objectId	string	是	否
TaskName	string	否	否

7. 任务记录实体表 (UserTask) 包含记录 Id (objectId)、对应的任务 Id (taskId)、对应用户 Id (user)、任务名称 (taskname)、任务完成标识 (complete)，如表 3-7 所示。

表 3-7 任务记录实体表

名称	数据类型	是否为主键	允许为空
objectId	string	是	否
taskId	string	否	否
user	string	否	否

续表 3-7:

名称	数据类型	是否为主键	允许为空
objectId	string	是	否
taskId	string	否	否
user	string	否	否

各表关系如图 3-8 所示:

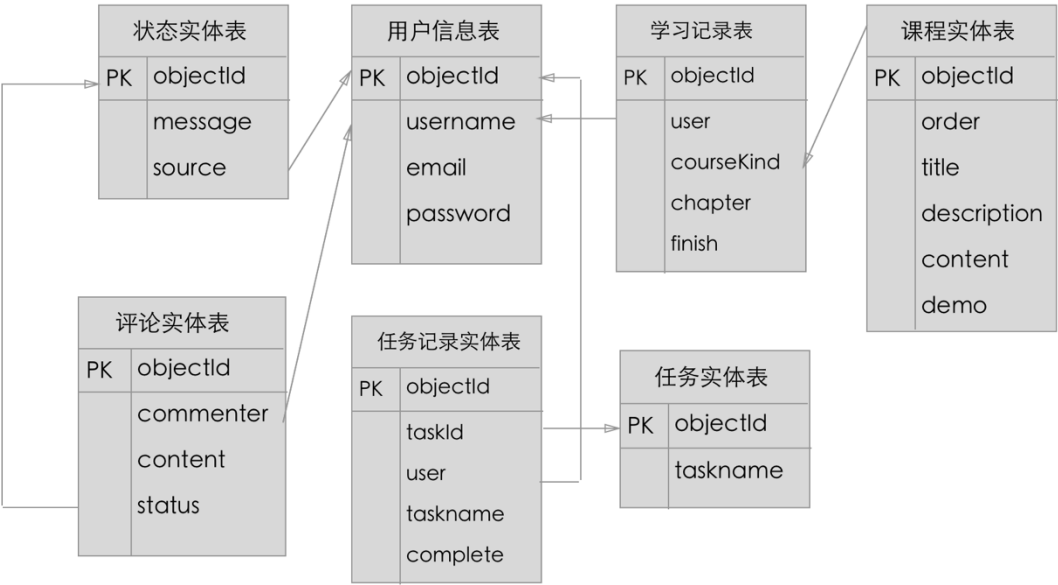


图 3-8 各表关系图

4. 系统详细实现

本网站主要由 10 个视图构成，分别是课程列表页，课程学习页，状态发布视图，状态列表，状态评论，学习进度，学习任务，登录，注册，修改密码。

首先介绍项目的结构以及个文件的作用。项目名为 `graduation-design`，包含后端依赖框架 `node_modules`，前端文件 `public`，后端入口文件 `app.js`，后端功能文件等。其中 `public` 文件还包含依赖框架 `materialize`，`js` 文件夹中还包含了框架 `LeanCloud` 的依赖包。项目目录结构如图 4-1 所示。

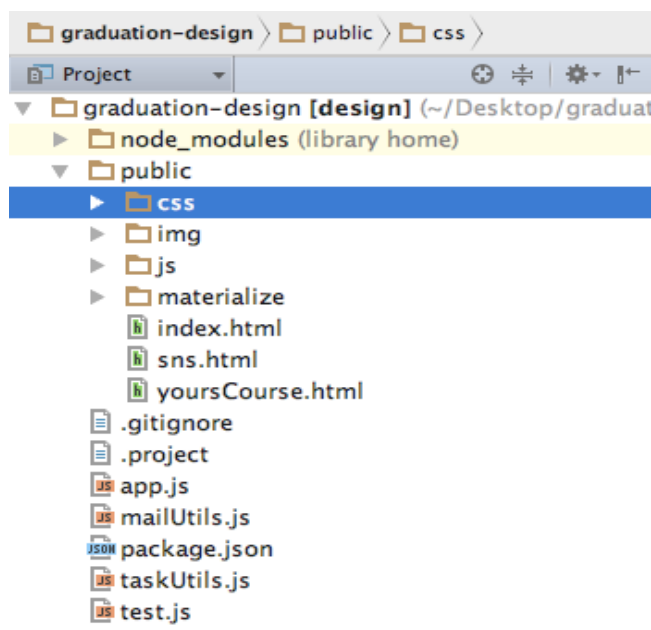


图 4-1 项目目录结构图

本网站为了提供良好的用户体验，在用户操作会产生反馈的地方多应用 `Materialize.toast()` 方法，以此来取代了浏览器原生的 `alert`。

4.1 后端实现

后端主要包含应用 LeanCloud 进行数据存储的云端服务和以 NodeJS Express 搭建的服务器组成。LeanCloud 存储网站所有数据，前端通过 JavaScript 与后端进行通信。而搭建的服务器则为任务学习功能提供了接口，用来提供任务信息，定时发送邮件等。后端会定时去遍历所有用户的任务完成情况，对未完成的用户进行邮件提醒，对已完成的用户发送奖励，从而更好地去驱动用户学习。

4.1.1 数据服务连接

首先在 HTML 文件中要先引入 LeanCloud 的依赖文件。

```
<script src="js/av-min.js"></script>
```

根据该项目应用的 App Key 与 App ID，在 JS 文件实现服务连接。

```
$(document).ready(function() {
    AV.initialize(
        'iuO5g66bCpCVIhnRtQnmn3YA-gzGzoHsz', 'xkRJahD7klcYeHQ3BDVbbwDS'
    );
});
```

4.1.2 数据库创建

以 Record 为例，创建 Record 表代码如下：

```
var Record = AV.Object.extend('Record');
```

在 LeanCloud 中，一个数据表相当于一个类，表中的每一条数据相当于一个对象。

4.1.3 数据库操作

数据的增改查应用 LeanCloud 特定方法，将在下文的用户模块中介绍。

4.1.4 网络请求

在用户查看学习任务情况的时候，浏览器向服务器端发送 GET 请求，请求任务详情，服务器查询数据表的相应任务，并且返回 JSON 数据给浏览器。

4.2 用户模块

用户模块分为用户注册、登录、修改密码、找回密码、退出登录五大功能。_User 类具备完善的属性与用户安全性。包含唯一标识 objectId、属性 username、email、password 和 emailVerified。

4.2.1 注册

用户点击导航栏右侧按钮开始注册操作，按要求填写相应字段。_User 表中所有属性不需要手动添加，在用户操作后，属性在逻辑层向数据库请求时自动生成，这不同于普通的数据库，需要提前设置属性。

注册功能涉及到数据的增加，数据的增加实质就是创建一个类的新对象，给对象的属性赋值之后保存该对象。具体实现如下。

```
var user = new AV.User();
user.set('username', userName);
user.set('password', userPassword);
user.set('email', userEmail);
user.signUp().then(function(user) {
}, function(error) {
    alert(error.message);
});
```

4.2.2 登录

在登录时，用户可以选择使用用户名或者密码进行登录，任何一个用户的用户名或者邮箱都是唯一的。登录代码如下：

```
AV.User.logIn(typeEmail, typePassword).then(function() {  
    location.reload();  
}, function() {  
    Materialize.toast("密码或用户名不正确！ ", 3000, 'rounded');  
});
```

登录弹窗如下图所示。

A login form with a light gray background. It contains two input fields: the first is labeled '用户名/邮箱' (Username/Email) and the second is labeled '密码' (Password). Below the password field is a blue link labeled '忘记密码?' (Forgot password?). At the bottom right is a green button labeled '登录' (Login).

图 4-2 登录图

用户可以输入用户名或邮箱登录，登录之后右上角区域将显示用户名。点击用户名，将出现下拉菜单。如图 4-3、4-4、4-5 所示。



图 4-3 导航 1

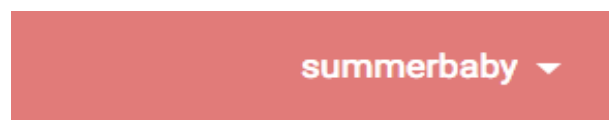


图 4-4 导航 2



图 4-5 用户模块图

4.2.3 找回密码

找回密码一般在用户登录的时候发生，所以在登录视图中。用户在输入框内填写注册时的邮箱，点击忘记密码，LeanCloud 会发送重置密码的邮件到用户邮箱。用户按提示进行操作即可。

重置功能涉及到数据库的查询，数据库的查询不仅可以使用方法，还可以利用 SQL 语句进行，具体可参照 LeanCloud 官方文档。实现如下：

首先查看相应的数据库：

```
var query = new AV.Query('_User');
```

然后，执行查询的条件匹配：

```
query.equalTo('email', $('#login_email').val());
```


接着对查询结果进行处理：

```
query.find().then(function(results) {  
    if (results.length == 0) {  
        Materialize.toast("邮箱未注册！请先注册", 3000, 'rounded');  
    } else if (results.length == 1) {  
        AV.User.requestPasswordReset($('#login_email').val()).then(function() {  
            Materialize.toast("重置链接发送成功，请注意查收邮件！", 3000, 'rounded');  
        });  
    }  
})
```

当用户点击忘记密码时，逻辑层会先行判断邮件格式是否正确，然后才执行发送重置密码的邮件。

重置密码邮件如图所示。

重设 learn-SNS 的密码 ☆

发件人: learn-SNS <do-not-reply@apps.leancloud.cn> 

时 间: 2016年5月10日(星期二) 下午4:45

收件人: 夏佳 <596804755@qq.com>

Hi, test

您请求重设应用 learn-SNS 的密码。点击下列链接来重设您的帐号密码（链接在 48 小时内有效）：

<https://leancloud.cn/dashboard/reset.html?token=XFasaeitpiywKOylx2ICArVShIF9bei8>

图 4-6 重置密码图

4.2.4 修改密码

在用户登录之后，才能对密码进行修改，要保证两次输入的新密码一致，不需要提供旧密码。

修改密码涉及到数据库的修改，数据库的修改就等同于先进行查询，找到相应对象后对其进行赋值。修改代码如下：

```
var user = AV.User.current();
user.setPassword(newpass);
user.save().then(function(){
    AV.User.logOut();
    Materialize.toast("密码修改成功！", 1000, 'rounded');
});
```

4.2.5 退出登录

用户如要切换账户，可以选择退出后重新登录。该功能的实现只需应用 `AV.User.logout()` 方法。但在页面加载时会去读取浏览器中的 `session` 信息，因此还需清除浏览器 `session` 记录，保证在同意浏览器重载页面不会出现无法退出的情况。

```
AV.User.logout();
sessionStorage.clear();
location.reload();
```

4.3 课程模块

课程模块主要由课程的展示和课程的学习组成。所有课程在首页展示，学习进度会跳转到新的课程页面，但课程的学习功能不会跳转，会在当前页面展示。

4.3.1 课程展示

课程的展示实现是由网页 DOM 加载完成后逻辑层去查找数据库中有哪些课程，然后呈现到页面上。用户选择相应课程时，绑定监听事件，并传递 `event` 参数，逻辑层通过点击事件获得目标课程 `event.target`，以该课程为条件，到数据库中找到与之匹配的课程数据库。根据表中 `chapter` 属性判断用户是否为第一次学习，若是课程学习页面显示第一章节的内容，否则显示用户上次学至的章节。为了课程进度存取方便，将这类的一部分信息临时保存在 Session 中。

首先取出在 Session 中的信息，

```
var kind = sessionStorage.getItem('kind');  
var chapterNu = sessionStorage.getItem(kind);
```

然后，查询 kind 表，

```
var query = new AV.Query(kind);
```

把课程的章节长度存入 Session，然后用 `get()` 方法取出表中相应的字段信息。

```
query.find().then(function(results) {  
    sessionStorage.courseLength = results.length;  
    // 处理返回的结果数据  
    var object = results[chapterNu - 1];  
    var order = object.get('order');  
    var title = object.get('title');  
    var content = object.get('content');  
    var des = object.get('description');  
    var demoString = object.get('demo');  
});
```

课程列表如下图所示。

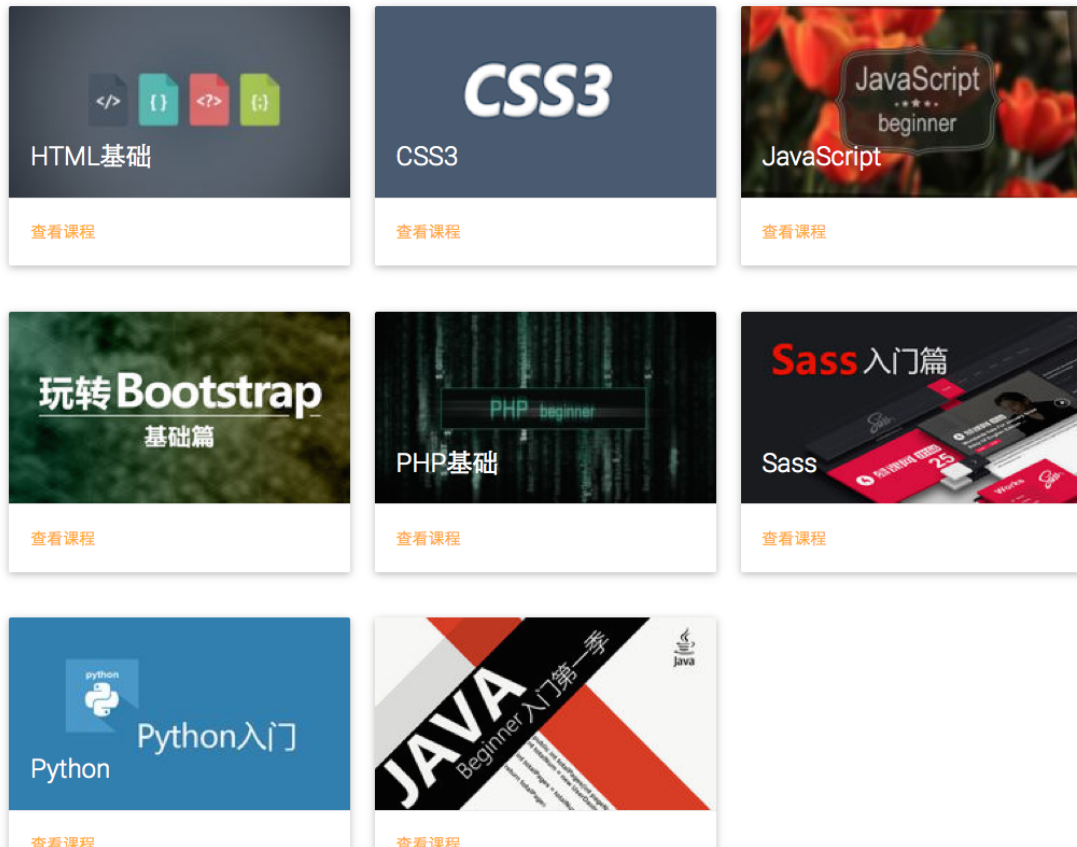


图 4-7 课程列表图

4.3.2 课程学习

用户进行学习时，通过点击上一节、下一节两个按钮来控制学习章节，逻辑层获得当前的章节数，并根据情况自加或自减，从而得到目标章节，此时逻辑层查询课程信息表，返回目标章节的具体内容到浏览器并且记录用户目前学习的进度。与此同时，还要修改该用户 Record 表中的章节信息，保证用户在进行下一次操作时请求到正确的数据。

首先得到当前的用户，

```
var currentUser = AV.User.current();
```

```
var kind = sessionStorage.getItem('kind');
```

接着创建学习记录表，向其中保存目前的学习进度。

```
var Record = AV.Object.extend('Record');
```

```
var record = new Record();
```

```
record.save({
  user: currentUser.get('username'),
  courseKind: kind,
}).then(function(record) {
```

```
sessionStorage.recordID = record.id;
}, function(err) {
});
```

课程学习图如下。



图 4-8 课程学习图

4.3.3 课程进度

用户点击进入课程中心可查看正在学习的课程与未学完的课程。由于用户访问该页面可能出现未登录的情况，因此学习进度的查看应首先判断用户是否登录。

课程进度信息的展示由逻辑层读取 Record 表中该用户的学习信息，由表中 finish 属性标识该课程用户是否学完，逻辑层判断之后返回两类数据给浏览器。浏览器得到数据之后，将相应的数据安排到指定的 DOM 元素上，接着把相应的 DOM 元素动态的添加到页面上去，同时把原来的内容清空。

正在学习的课程与已学完的课程如下图所示。

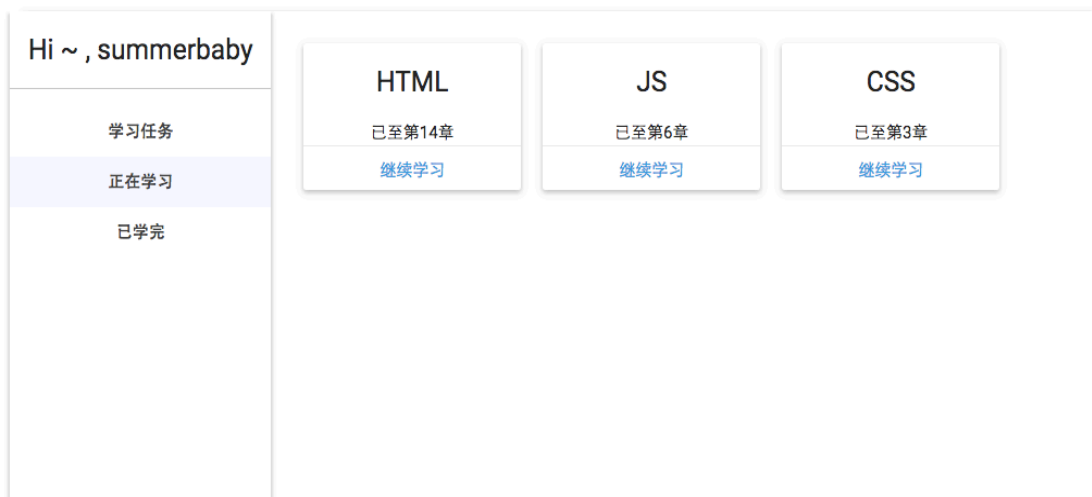


图 4-9 已学课程图



图 4-10 已完成课程图

4.4 社区模块

4.4.1 发布状态

用户进入社区之后可以自由发布状态，但发表内容不能为空。该功能实现首先在前端页面构建发表状态布局，在用户点击发布之后，逻辑层捕捉到点击事件，获取用户 Id 和用户发表内容，将数据存入云端数据库_Status，则状态发布成功。

发表状态视图如图 4-11 所示。



图 4-11 状态发表图

4.4.2 状态列表

用户想要查看状态列表，只需要点击公开状态按钮，逻辑层监听到点击事件之后，到云端数据库查询 `_Status` 表，将其中状态对象逐条地返回到浏览器状态列表视图之中。

公开状态列表如图 4-12 所示。

posttest	2016-5-14 17:12	+ 关注
求推荐前端学习书籍~~		
查看评论		
Second	2016-5-12 20:53	+ 关注
大家学习html/css是直接在这里看，还是还买书看？		
查看评论		
summerbaby	2016-5-12 20:46	取消关注
HTML5哪个编辑器好？		
查看评论		
summerbaby	2016-5-12 20:46	取消关注
前端与后端哪个更难？		
查看评论		

图 4-12 状态列表图

4.4.3 状态评论

每一条状态下方都有查看评论按钮，用户点击之后，页面会弹出评论详情，逻辑层查询 `Comment` 表中该状态的评论，逐条取出渲染到前端页面中。用户可以浏览该状态下不同用户的评论，也可以发表自己的评论。只需填写评论内容，点击评论按钮，逻辑层将用户评论内容，用户 `Id` 等存入 `_Status` 表中，则完成了评论提交。

评论列表如图 4-13 所示。

Second	2016-5-12 20:52
推荐Github出品的Atom,免费,而且有智能易用的代码提示,还能查询标签信息,有或者是Bluefish Editor	
test	2016-5-12 20:49
Sublime	
test	2016-5-12 20:49
Hbuilder	

图 4-13 评论列表图

添加评论如图 4-14 所示。



图 4-14 添加评论图

4.4.4 朋友圈

用户可以在公开状态列表上选择关注他人或关注后取消对他人的关注。并且点击朋友圈，则可以查看已经关注过的朋友的状态。在导航栏走上方点击关注详情按钮，查看我的粉丝和我关注的人。

朋友圈状态如下图。

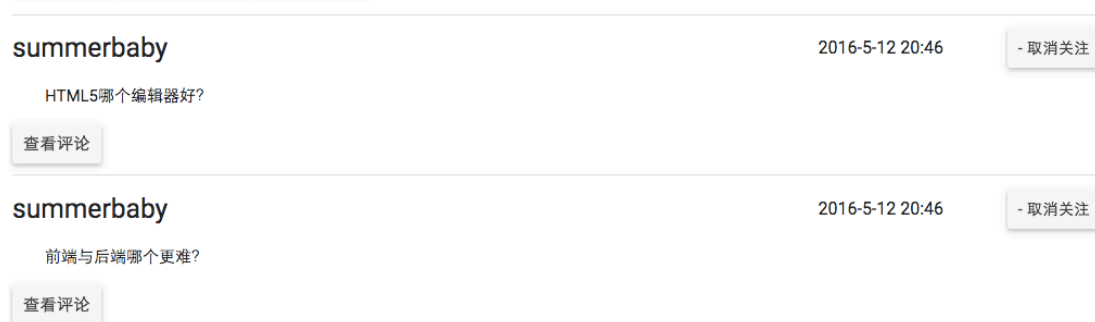


图 4-15 朋友圈状态图

在导航栏右方点击可查看关注详情，如图 4-16、4-17、4-18 所示。



图 4-16 关注详情按钮图



图 4-17 关注详情图

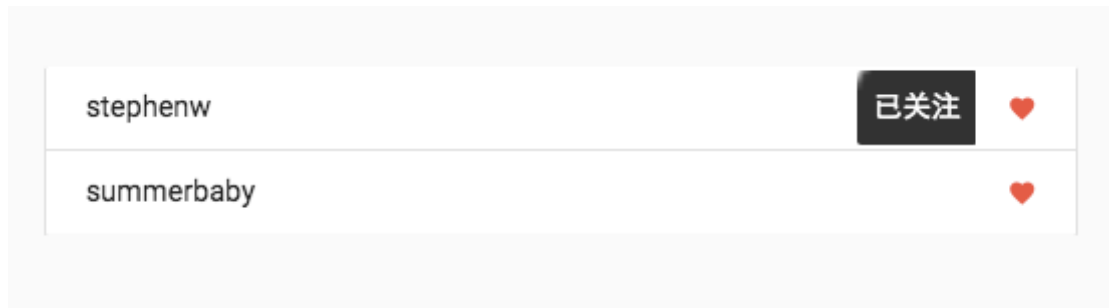


图 4-18 已关注图

4.5 任务模块

4.5.1 任务列表

用户在个人中心可以查看到任务列表，其中包含了任务的具体信息、用户的完成情况。任务数据由逻辑层向服务器发送 GET 请求，请求 tasks 接口，服务器到云端数据库查询任务信息，返回 JSON 格式的数据，由逻辑层进行判断任务是否完成，从而决定每条任务详情最右端存在的图案标识的类型，并且利用 tooltip 方法，实现鼠标在图案上悬浮时的文字提示，提升用户体验。

```
var HTMLString = " <ul class='collection'>";
$.ajax({
    type: "GET",
    url: "/tasks",
    data: {userid: currentUser.id},
    dataType: "json",
    success: function(data){
        console.log(data);
        data.forEach(function(t){
            if (t.complete) {
                HTMLString += "<li class='collection-
item'>"+t.taskname+"<i class='mdi-action-done small green-text tooltiped' data-position='left'
```

```

data-delay='50' data-tooltip='已完成'></i></li>";
        } else {
            HTMLString += "<li class='collection-
item>'"+t.taskname+"<i class='mdi-action-schedule small red-text tooltiped' data-position='left'
data-delay='50' data-tooltip='未完成'></i></li>";
        }
    });
    HTMLString += ' </ul>';
    $('all_course ul').append(HTMLString);
    $('tooltiped').tooltip({delay: 50});
}

});

```

效果图如下：

Hi ~ , summerbaby	<div>完成10节HTML课程</div> <div>已完成 </div>
学习任务	<div>完成全部JavaScript课程</div> <div></div>
正在学习	<div>完成全部HTML课程</div> <div></div>
已学完	<div>完成5节JavaScript课程</div> <div></div>

图 4-19 学习任务图

4.5.2 任务提醒

任务提醒由服务器端实现，服务器端会定时遍历 Usertask 表，根据 complete 参数判断用户任务完成情况。对未完成任务的用户即时发送邮件提醒。同时，也可视情况，为完成的用户发送奖励，达到驱动用户积极学习的目的。

设置发送邮件的服务器：

```

var mailgun = require('mailgun-js')({apiKey: api_key, domain: domain});
var mailUtils = {
    send: function(message,receiver,callback){
        var data = {
            from: "noreply@stephenw.cc",
            to: receiver,

```

```

    subject: message.subject,
    text: message.text
  };
  mailgun.messages().send(data, function(err, body){
    callback(err, body);
  });
}
};

```

module.exports = mailUtils;

提醒邮件实现效果如下：

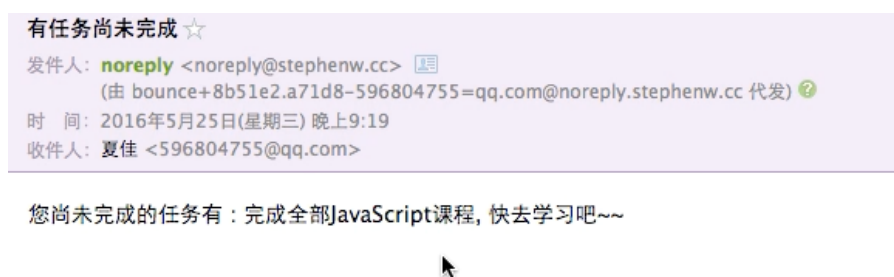


图 4-20 任务提醒图

5. 系统测试

在网站开发完成之后，需要对网站进行多种测试。测试结果表明，网站所有功能运行良好，界面无明显 bug。

表 5-1 测试结果表

测试功能	运行情况	测试时间
首页	是	2016-5-1
登录	是	2016-5-1
注册	是	2016-5-1
退出	是	2016-5-1
修改密码	是	2016-5-1
找回密码	是	2016-5-1
课程学习	是	2016-5-10
正在学习课程	是	2016-5-10
已学完课程	是	2016-5-10
学习任务	是	2016-5-10

续表 5-1:

状态列表	是	2016-5-12
状态发布	是	2016-5-14
状态评论	是	2016-5-14

结束语

至此，本网站的设计与开发工作圆满结束，经过 Chrome、Safari、Firefox 等多个浏览器测试，本网站兼容性良好，各个模块都能按照预期正常运行。在这次技术学习平台网站的开发过程中，我对以往所学的知识进行了回顾与总结，把书本上的知识运用到了实践之中，对之前比较模糊的知识点有了比较清晰深刻的认识，并且能够熟练地运用起来。我对页面布局与重构有了深入的掌握，对 Compass 与 Sass 的原理与使用有了一定的了解，清楚了网页常用的加载图片的方式，通过对前后端通信的处理，我能熟练地用 JavaScript 处理逻辑，学会用了 NodeJS Express 搭建后台。在此过程中，我深刻地体会到书本上的知识只能留在书本上，只有用手去做去写去研究，才能切实提高自己能力的道理。

本网站基于 Materialize 前端框架，该框架应用 Google 的 Material Design 设计风格，简洁美观。同时网站应用 Leancloud 的 BaaS 框架做数据存储，使用 NodeJS Express 搭建服务器。由于需要设置邮件分发、任务提醒功能，所以需要配置服务器来定时推送邮件。

由于个人的知识有限，开发经验存在不足，本网站还有一些可以优化和提升的地方。例如：

1. 课程搜索：目前网站只提供了课程列表，没有提供课程搜索功能。搜索能够让用户更快的找到要学习的课程。
2. 资源不够丰富：由于现实条件的限制，网站的学习资源还不够丰富，教学形式比较单一，如果条件允许，可以考虑引入视频资源等。
3. 社区模块状态列表暂不显示发表者的用户名，由于显示一条就去需要去请求一次数据库，会造成浏览器负担过大。

另外，由于开发时间有限等种种原因，本网站代码还不够精简，部分重复的 DOM 结构没有实现模板利用，最开始的有一些构想，最后也没能实现，不得不说这非常地遗憾。这是我第一次独立开发一个网站，从设计风格到布局实现再到后端搭建，多少都会存在一些不足，希望各位老师批评指正。希望在日后能够继续地优化它的结构与功能，让它真正能够去帮助 IT 初学者快速入门，早日成为技术大牛。

参考文献

- [1] L Pappano. The Year of the MOOC[N]. The New York Times, 2012-11-2.
- [2] Li Nan. MOOC Learning in Spontaneous Study Groups: Does Synchronously Watching Videos Make a Difference [R]. Switzerland: P.A.U. Education, 2014-1-2.
- [3] Eric T Freeman. Head First HTML With CSS & XHTML[M]. O'Reilly Media. 2005-12-18.
- [4] Micheal Nanfito. MOOCs: Opportunities, Impacts, and Challenges[M]. CreateSpace Independent Publishing Platform. 2013-12.
- [5] Ethan Brown. Web development with Node and Express[M]. O'Reilly. 2014-7-25.
- [6] Douglas Crockford. JavaScript 语言精粹[M]. 赵泽欣 / 鄢学鹏 译. 电子工业出版社. 2009-1.
- [7] David Flanagan. JavaScript 权威指南[M]. 张铭泽 译. 机械工业出版社. 2003-1-1.
- [8] Ben Frain. 响应式 Web 设计[M]. 王永强 译. 人民邮电出版社. 2013-1-1.
- [9] Mike Cantelon. Node.js In Action[M]. 人民邮电出版社. 2014-5.
- [10] Ben Frain. Sass 和 Compass 设计师指南[M]. 任凯光. 华中科技大学出版社. 2014-4-25.
- [11] 廖伟华. 图解 CSS3[M]. 机械工业出版社. 2014-7-1.
- [12] 弗拉纳根. JavaScript 权威指南[M]. 李强 译. 北京：机械工业出版社. 2007 年.
- [13] Chuck Musciano. HTML 和 XHTML 权威指南[M]. 技桥 译. 北京：清华大学出版社. 2015 年.
- [14] 罗宾斯. Web 前端工程师修炼之道[M]. 谢合亮 译. 机械工业出版社. 2014-9-1.
- [15] Jeremy Keith. JavaScript DOM 编程艺术. 杨涛 译. 人民邮电出版社. 2011-4.

致谢

衷心感谢我的导师王定成老师，在系统设计和论文修改过程中给我提供了非常耐心的指导和帮助。

感谢心动网络平台事业部所有同事给我提供的 UED、技术以及架构上的指导和帮助。

感谢王志龙对后端架构和 NodeJS 技术学习的指导。

最后感谢学校和各位领导能给我们提供外出实习的机会。感谢答辩老师！感谢学院！