

# 课程管理网站 MyCampus 项目个人总结

517070910133 阮颖

2018 年 12 月 18 日

## 1 项目概况

### 1.1 项目初衷

大学生在学习过程中，每个学期的课程都会不一样，而且每门课程的学习资源非常分散，在传统的教学信息服务网上面只能得到每门课的上课时间，地点及任课教师，这些信息对于学生们来说是远远不够的，像老师布置的课程作业以及相关课程的学习参考资料我们无处得知。所以项目的目的就是搭建这样的一个课程管理网站，帮助我们管理好学期内的每一门课程，让我们能够在这个网站上获取跟课程有关的所有信息。

### 1.2 项目整体规划

虽然不是项目的 master，但是我还是想说一说我们组对于整个项目的工作安排，因为在我看来清晰细致的工作安排让我们整个团队的工作有条不紊地进行了下去，这是我们团队的一大亮点。在第一次会议中，经过讨论我们决定采用前后端分离的方法，以“前端先行完成网站设计并提出功能需求，后端再逐步实现功能跟进”为开发原则。所以我们被分为前后端两个小组，组长任命了前后端的小组长，而小组长负责任务分配以及前后端比较大的交接。

第一阶段：首先是我们前端用 HTML 快速开发出最原始的界面并且尽可能地完成 CSS 和 JavaScript 代码，这个界面包含了大部分后端需要实现的需求。

第二阶段：接下来后端就可以与前端进行交接了。前端在注释中指明需要后端链入数据库的部分，再由后端修改代码来实现网站的需求。同时前端配合后端和继续美化界面。

第三阶段：后期的工作以后端为主，前端为辅。后端需要完成数据库、网站的跳转以及服务器搭建配置、上传网站等工作，前端完善界面，加入一些新的功能。

## 2 个人工作总结

我在该项目中的工作是负责网站前端的 account 界面和 help 界面，并配合后端同学对页面代码进行改写。

### 2.1 第一阶段

经过前端小组的讨论，课程管理网站由 login、home、course、assignment、inbox、account 和 help 共 6 个界面构成，而我负责的则是 account 和 help 两个界面。

在之前我虽然写过一些代码，但都是在学习某门编程语言的过程中写的十分简短的小程序，一般不会超过 50 行，而像这种开发一个完整的项目，需要团队协作的，我是第一次做，没有任何的经验。这样无疑给我带来了很多的困惑：前端语言 HTML、CSS 和 JavaScript 以前都没有接触过，也不知道如何去布局页面。所以刚开始的时候我的工作开展起来非常困难。看着书架上那本 600 多页的《HTML5+CSS+JavaScript 从入门到精通》，不禁犯了难，短短一个月的时间看完这么一本书是完全不可能的。所以我大概地梳理了这本大部头的结构，挑出了里面最核心的几个部分进行快速地学习，掌握

HTML 和 CSS 中最核心的东西。然后访问一些界面比较好的网站，用 Google Chrome 去学习别人优秀的源代码以及别人的编程规范。就这样大概学了两周左右，我确实能写出来一点像样的 HTML 和 CSS 代码了。

搭建好前端环境之后，我们前端就开始讨论网站设计的各种细节，每一个标签使用什么样的 CSS 样式，怎样搭建出简洁而有美观的界面，等都是我们讨论的内容。

在我所做的部分中，account 界面能够显示用户的姓名、学号、学院、专业等信息，这一部分是需要导入后端的用户数据库的，而为了提高效率，我先将 account 页面写成一个 HTML 文件，在需要导入数据库的地方加上注释，方便后期修改。

```
<div id="content-outer"><div id="content-wrapper" class="container_16">

  <!-- main -->
  <div id="main" class="grid_8">
    <h2>Personal Information:</h2>

    <!-- 此处需要导入学生信息-->
    <!-- 导入姓名-->
    <h4>Name:   ***</h4>
    <!-- 导入学号-->
    <h4>Student Number:   ***</h4>
    <!-- 导入学院-->
    <h4>Academy: School of Electronic Information and Electrical Engineering </h4>
    <!-- 导入专业-->
    <h4>Major:   Information Safety</h4>
  </div>
</div>
</div>
```

图 1: account.html

而 help 界面相对来说比较简单，这个界面相当于一个帮助文档，指导网站的用户去怎么使用网站。

```
<!-- content starts -->
<div id="content-outer"><div id="content-wrapper" class="container_16">

  <!-- main -->
  <div id="main" class="grid_8">
    <h1>Help</h1>
    <h3>Login in/Login up</h3>
    <ul>
      <li>第一次使用该网站时需要注册，注册需要提供id（用户名），密码，姓名，学号，学院，专业。</li>
    </ul>
    <h3>HOME</h3>
    <ul>
      <li>包含您的全部课程的工具栏(Courses)，点击相应的课程链接可以跳转到相应的课程界面。</li>
      <li>中间栏是您的全部通知，敬请关注！</li>
      <li>右侧栏(To-Do-List)包括全部课程的任务，点击相应课程链接可跳转到相应的任务界面。</li>
    </ul>
  </div>
</div>
```

图 2: help.html

CSS 样式部分是我们前端小组共同商讨，共同完成的，这样我们就可以在 HTML 中引用相对应的样式。

```
body {  
    font: 12px/165% 'Lucida Grande', Verdana, Helvetica, sans-serif;  
    color: #666666;  
    margin: 0; padding: 0 0 50px 0;  
    background: #F6F5F5 url(../images/bg.gif) repeat-x;  
    text-align: center;  
}  
/* Links */  
a:link, a:visited {  
    text-decoration: none;  
    color: #7ba857;  
}  
a:hover {  
    border-bottom: 1px dotted #7ba857;  
}
```

图 3: KeepItSimple.css

## 2.2 第二阶段

其实在第一阶段，前端的工作量是比较大的，既要学习前端语言，又要搭建出最初的界面。第二阶段是整个项目开发过程中进展最艰难的阶段，后端要实现我们提出的功能，前端还要根据实际实现情况改变网站的结构，所以我们在这之中阉割了不少我们暂时无法实现的功能。

而我在这个阶段屡屡碰壁。因为我们的静态网页需要链接数据库，需要安装后端的开发环境，我们又安装了 PHPStorm。但是光配置 PHP 环境就花了我很长的时间，按照网上教程的配置一步步来，却还是无法运行，之后就在 Google 上搜索报错信息，看了 Stackoverflow 上面的数十个解答才发现了问题所在，最终成功地运行了我们的代码。经过这么一番折腾，我终于认识到 Stackoverflow 是一个多么强大的网站，在上面也确实学到了很多遇到问题的解决办法。

之后我又粗略地了解了数据库语言 MySQL, PHP 的知识，改写了部分代码，让我们的界面能够链接数据库。

## 2.3 第三阶段

第三阶段其实算是我们的收尾工作了，将我们的代码、数据库等放在服务器上试运行，但是这看似简单的工作，恰恰是技术含量比较高的部分，各种 bug 都在这一阶段出现了。但是由于个人能力原因，再加上我不了解后端以及服务器的知识，所以只好在完成自己的界面后构思加入一些新奇实用的小功能让我们的网站看起来更酷。

# 3 个人收获

说实话，通过这门软件工程，或者说通过这个网站的开发，我确实学到了很多实用的东西，比如编码规范，Github 的使用以及各种 IDE 的使用。我相信这在其他课程如 C++ 中是无法学到的。

### 3.1 代码规范

大一的时候学习 C++，并没有很重视代码的规范性，变量的命名从来都是英文字母，函数的命名也比较地混乱，不能从命名看出一个简单的程序想干什么；那时注释对于我来说也是可有可无的，最多也只是写明一个几十行的程序的功能。而在项目的开发过程中，我发现这种做法是多么的低效，如果我在我的 HTML 文件中没有用注释指明需要链入数据库的部分，后端的同学估计要花很长的时间来读懂我的代码，然后才能找到需要导入数据库的代码吧。所以说代码规范在团队开发中起着相当重要的作用。代码规范还包括了源文件如何命名，变量怎么命名，怎么缩进以及文件怎么分开管理等内容，自己以后在编码过程中会多多注意编码规范问题，让自己的代码能够让其他人很容易看懂。

### 3.2 Github+Markdown

在上这门课之前，我是听说过 GitHub 这么一个网站，但是并不知道它可以用来干嘛，后来经过老师的讲解才明白这个网站的作用，后来在开发的过程中，发现 WebStorm 和 PHPStorm 可以通过 Version Control 直接修改 Github 上的代码，这确实给我们的开发带来了很大的便利。后来又通过实践探索了 Github 很多其它强大的功能，现在我也会会把自己写的一些 demo 放在 Github 上，让其他人可以纠正和改进我的代码，也算是一种对自己学习成果的一种肯定吧。

之前我并不知道 markdown 这种标记语言，只知道写论文常用的 LaTeX。老师要求提交自己的 idea 时，当我看到其他同学提交的文件在 Github 上显示的效果之后，惊呼：这是什么文件，排版真的太好看了！经过了后来的学习，深入了解了 Markdown 语言，发现它如此的简单又如此的实用，确实是我们这些程序员的一个利器。

### 3.3 Tutorial+Stackoverflow

之前我想自学 Python 的时候，想在网上搜索一些现成的中文教程，总是会因为大多数教程不够完整而且存在大量的纰漏而苦恼，所以学习的 Python 知识也就是零零碎碎的，无法形成一个完整的体系这大大地降低了我的编程兴趣。通过这门课了解到了官方文档的重要性，学习一门语言就要去看它的官方文档，不仅免费，而且比绝大部分教程全面易理解。比如现在我学习 Java 就会去看它的官方文档，这样不仅可以系统地学习，还可以提高自己的英文水平。

自从知道了 Stackoverflow 这个网站之后，我真的想说：这个网站真的是程序员心中最好的网站。当我的程序出现莫名其妙的 bug 后，在这个网站搜索，准能搜到想要的答案，还能了解这个 bug 出现的原因。

官方文档再加上 Stackoverflow 这样的一种学习计算机知识的方法是高效而且深刻的，它能让我在所学东西的广度和深度上都有很大的提升，我以后会利用这些来达到最高的效率。

## 4 总结

总之我觉得这门课的意义非凡，它更像是国外的那种课程，更能够让我们学到很多实用的东西而不仅仅是理论知识。我衷心地希望以后能接触更多这样的课程，在实践中提升自己的能力。