

# 回顾与作品集

现在大三了，回顾了一下现在的自己和大一刚进入校门的自己，确实改变了很多也进步了很多，比起高中里带着功利性质的应试，我还是更加喜欢大学可以自由自在学习的氛围。记得大一刚进学校的时候我是个连软件都装不好的憨憨，万幸的是我还着有一颗想要进步的心，不至于终日沉溺在旧的舒适圈里不愿意走出来。

## 最最开始

相信很多朋友和我一样最开始对于编程的学习是从C语言开始的，学完之后写了一个比较简单的学生信息管理系统进一步锻炼自己的编程能力，代码链接放在最下面。

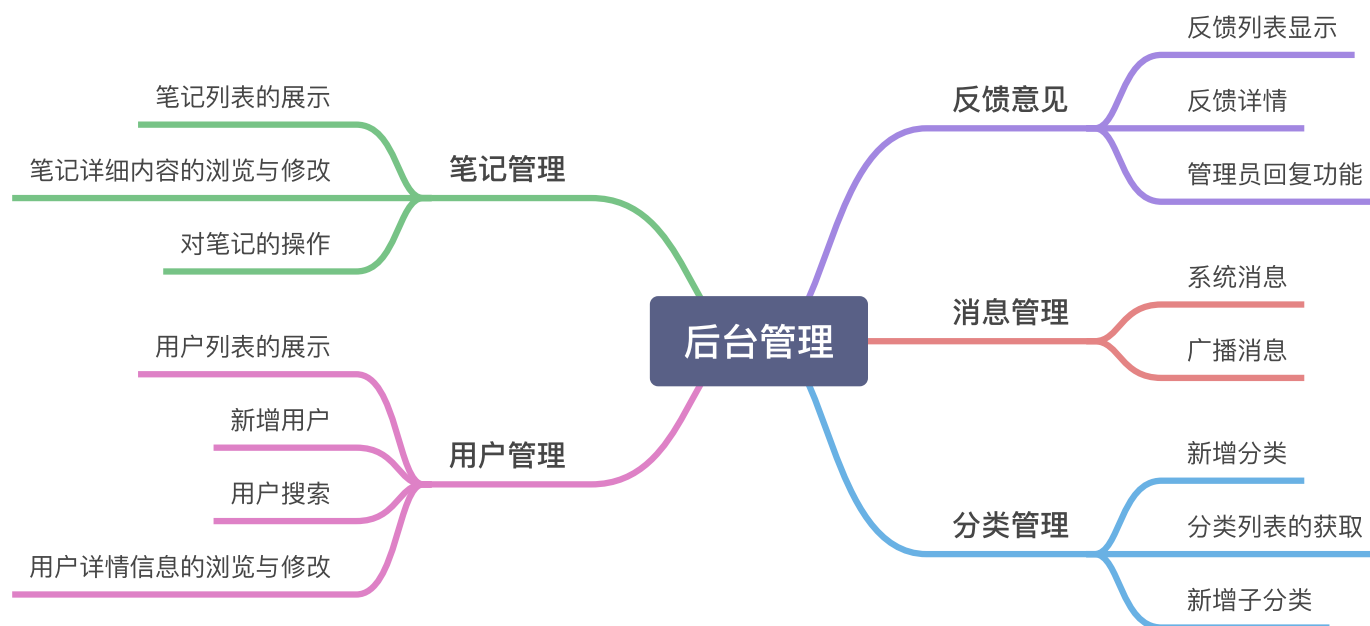
[Github 学生信息管理系统](#)

在大一的寒假里，受周围同学们的影响，我还去玩了单片机，也是用C语言编写，自己做过一个简易密码锁，这个过程还是很有趣的，让我定位问题的能力和解决问题的能力得到了提高。不过很可惜的是，当时做完就觉得完事儿了，没有把代码备份存起来。

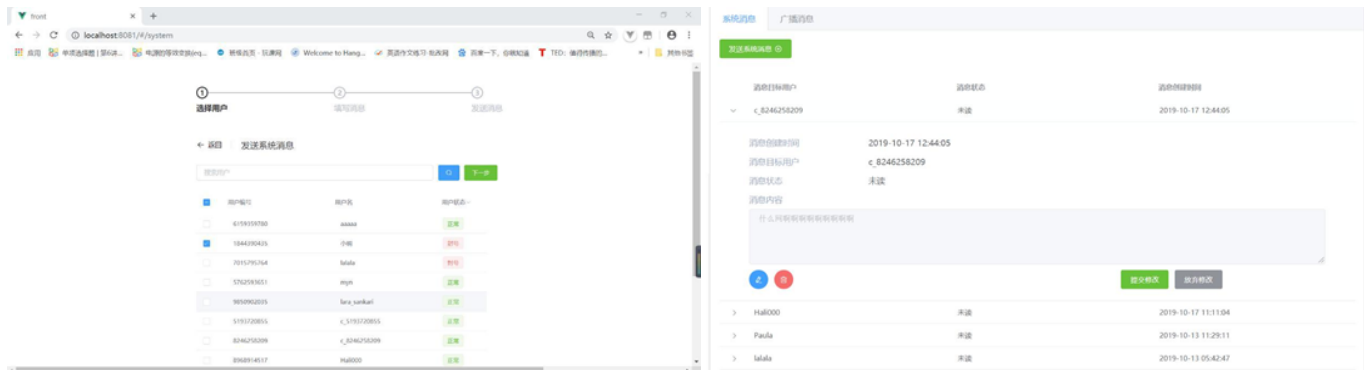
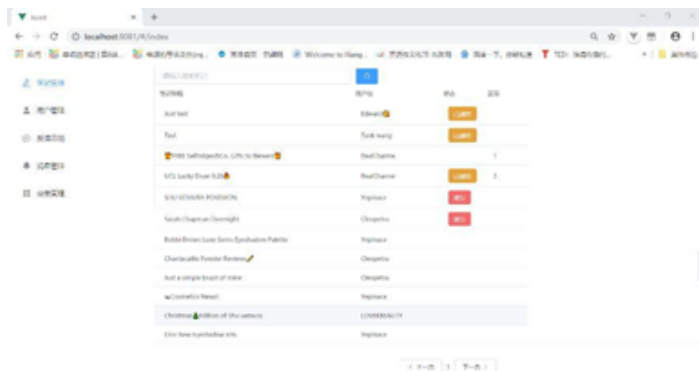
## web开发中做过的东西

大学里零零碎碎加起来也做过不少东西了

### Vue-cli +element UI搭建一个后台管理系统



一下是部分页面的展示，可能图片有点不清晰.... 唉



## 竞赛HELP小程序开发

我和我队友一开始做这个小程序的想法的来源是，学校里有很多的竞赛，像数学建模，大创还有省新苗之类的，这些竞赛呢有的可能需要跨学院组队，然后我们就希望能做一个小程序来发布一些竞赛相关的信息以及帮助同学们组队。

这个小程序是我在大二的时候做的，当时可能刚刚接触前端，然后技术上可能技术亮点没有多少，不过对于我当时一个前端小菜鸟来说通过做这个小程序也学到了不少东西的，比如说前后端分离的项目要怎么和后端协作沟通定义接口啊，用express搭一个简易的后端服务啊。因为小程序是获取接口必须要用https的，然后也了解了一些https方面的东西。再比如说一些公共的样式可以定义成一个模板template来达到一些样式的复用啊。

[Github 竞赛HELP小程序开发](#)

以下是一些展示啦：

竞赛HELP



2019年浙江省大学生科技  
创新活动计划暨新苗人才计划



2019年国家级大学生创新  
创业训练计划



2019年浙江省大学生科技  
创新活动计划暨新苗人才计划



2019年浙江省大学生科技  
创新活动计划暨新苗人才计划



首页



队伍



组队



我的

竞赛HELP



搜索team



比赛: 全国数学建模大赛

队名: 憨憨队 已审核



比赛: 全国数学建模大赛

队名: 憨憨队 已审核



比赛: 全国数学建模大赛

队名: 憨憨队 已审核



比赛: 全国数学建模大赛

队名: 憨憨队 已审核



首页



队伍



组队



我的

4G 2.1K/s 16:44 HD 4G 58%

竞赛HELP

比赛名称 全国大学生数学... >

队伍名称

队伍目前人数

队伍期望人数

队伍需求

队伍简介

联系人微信

提交



4G 1K/s 16:43 HD 4G 58%

竞赛HELP

搜索team

比赛: 全国数学建模大赛  
队名: 憨憨队 已审核

比赛: 全国数学建模大赛  
队名: 憨憨队 已审核

比赛: 全国数学建模大赛  
队名: 憨憨队 已审核

比赛: 全国数学建模大赛  
队名: 憨憨队 已审核

4G 0.1K/s 16:43 HD 4G 58%

阿毛

我创建的队伍 >

我加入的队伍 >

消息管理 >

我的个人信息 >

关于比赛助手 >





首页



队伍



组队



我的

## 使用Sails.js框架开发书城的服务端

大二下的时候老师给我推荐了这个框架，这是一个基于express的开发快速开发服务端的框架，可以开发面向数据的web应用程序，不足是没有中文文档，学起来需要自己去看英文文档，一开始可能有些累，不过由于这个框架使用起来还是十分简单的,所以最终还是做了个东西出来，对新东西的学习的能力也提升了。

学完之后做了一个书城项目，前后端分离，我负责后端提供接口，另一个小伙伴负责前端

[官方文档 Sails.js](#)

[Github 利用sails.js搭建书城项目](#)

## 使用React开发个人博客

### Important Point:

- 使用一个第三方模块，**styled-components**，把css写入js文件里作统一管理，同时也可以解决引入样式命名冲突的问题
- 使用redux来进行数据层的数据管理，解决多个组件可能要共用数据以及兄弟的兄弟组件之间需要传值的问题
- 一个reducer里面有很多代码可能会不好维护，所以把一个reducer拆分成很多个reducer。具体操作如下，比如有一个header组件，可以在header组件的文件夹下新建一个store文件夹，把header有关的reducer放到该文件夹下，存放header相关的默认数据和操作，然后再在总的reducer里面对所有的数据进行整合，通过combineReducers这个方法
- 引入了immutable.js,有的时候不知道咋肥四可能会把state给改了，为了解决这个问题，我们生成这个immutable（不可改变的）对象,immutable库里有一个fromJS对象，通过这个对象可以把js对象转换成immutable对象。
- 把所有的异步请求放在一起，千万不要写在组件里，后期维护会很累呢
- 对返回顶部的那个组件做了性能优化，加入了防抖函数
- 绑定的事件千万不能忘记在组件willUnmount的时候解绑

