**南开大学软件学院**

**企业实训总结报告**



|  |  |
| --- | --- |
| **项 目 名 称：** | 在线围棋教育平台 |
| **合作培养企业：** | 计蒜客 |
| **实 训 方 向：** | Web开发 |
| **姓 名：** | 魏鹏 |
| **实训起止日期：** | 2018.07.17-2018.8.31 |
|  |  |

**一 引言**

1.1项目概述

1.项目目的：为线上围棋教育平台提供报名和课程管理的线上系统

2.项目背景：在线围棋教育

3.项目意义：改善已有的教务和报名系统

4.参考资料：python官方文档与github的开源项目

1.2 需求分析

a.有在⽹网⻚页与⼿手机端可以访问的招⽣生落地⻚页⾯面，家⻓长可以在线填写信息并缴费

b.需要有管理理后台，后台⽀支持管理理在线填写线索的管理理、导⼊入第三⽅方来源线索

c.后台需要⽀支持课程顾问查看和跟进客户，并对报班、不不报班(及原因)进⾏行行标记

d.后台需要可以查看正在上课的班级，并⽀支持教务⽼老老师在上课后对已经上课进⾏行行销课

e.需要有完备的给到家⻓长、线索管理理者、课程顾问、教务⽼老老师、超级管理理员的⽂文档

f.对已经上完的课程可以进⾏行行存档，⽅方便便区分正在上和已经结束课程

d.数据可视化，⽅方便便精细化运营1.3 运行环境

**二 项目设计**

2.1 调研选型

**Django**

1．基于Django原生的admin自主管理后台，直接对应项目需求中的用户管理系统，并且其中原生的auth模块，可以直接实现登陆认证等，同时，auth模块中已包含密码加密的哈希算法，在应用上较为简便。

2．Django架构相对复杂，但同时相对清晰，开发中相互影响较小，适合对于架构理解并不深入的我们进行开发，降低了框架理解难度。

3．之前作业中有利用Django制作留言板，组内成员均对Django的原理有相对较深的理解，使用Django可以一定程度上减少重新学习一种框架的时间。

4．Django相对于Flask使用人数较多，文档较全，适合解决开发中遇到的各问题。

5．Django支持关系型数据库容易，且上学期开设数据库课程中主要为关系型数据库，应用较为简单。

**vue-resource**

1.因为vue.js本身是数据驱动，这使得我们并不需要直接操作DOM，如果我们不需要使用jQuery的DOM选择器，因此也就没有必要引入jQuery。

2.vue-resource是Vue.js的一款插件，它可以通过XMLHttpRequest或JSONP发起请求并处理响应。也就是说，$.ajax能做的事情，vue-resource插件一样也能做到，而且vue-resource的API更为简洁。

3.jQuery本身体积较大，而vue-resource体积上远小于jQuery，因此选择了vue-resource

4.并且vue-resource使用时间较长，使用范围也较广，人数居多，便于开发时问题的解决。

2.2 模块功能介绍

招生系统，课程顾问系统，教务系统，超级管理员系统

2.3 用户故事

1. 名称：注册试听课：

事件：

* 1. 用户根据链接注册
  2. 系统返回课程缴费界面
  3. 用户进行试听课缴费
  4. 系统根据信息进行存档

1. 名称：完善用户信息

事件：

* 1. 课程顾问查询用户
  2. 系统返回查询的用户的信息条目
  3. 课程顾问根据询问的信息进行完善

1. 名称：购买常规课

事件：

* 1. 用户进入购课页面
  2. 系统显示相应推荐课程
  3. 用户选择某项课程
  4. 系统进入支付页面
  5. 用户完成支付

1. 名称：排课

事件：

* 1. 教务老师进入排课页面
  2. 系统显示课程列表
  3. 教务老师选择具体课程
  4. 系统显示课程相关信息
  5. 教务老师根据用户选择的时间添加相应时间段课程

1. 名称：获取和修改个人信息

事件：

* 1. 用户进入个人信息页面
  2. 系统显示当前用户信息
  3. 用户进入修改页面进行修改
  4. 系统显示修改后的信息

1. 名称：教务老师销课

事件：

1. 教务老师查询待销课程
2. 系统显示待销课程
3. 教务老师选择销课
4. 系统提示销课成功
5. 名称：顾客退费

事件：

1. 顾客提交退费申请
2. 工作人员收到退费申请通知
3. 工作人员接受或拒绝申请
4. 顾客收到退费或退费取消通知

八：

名称：查询数据

事件：

1.超级管理员点击页面按钮进行查询操作

2.系统显示可选历史时间和数据种类

3.超级管理员点击选择某个时间某个数据种类

4.系统给出对应所需信息

九：

名称：调整运营策略

事件：

1.超级管理员点击页面按钮进行调整运营策略操作

2.系统显示可选删课、增课和更改价格操作

3.超级管理员点击选择操作种类

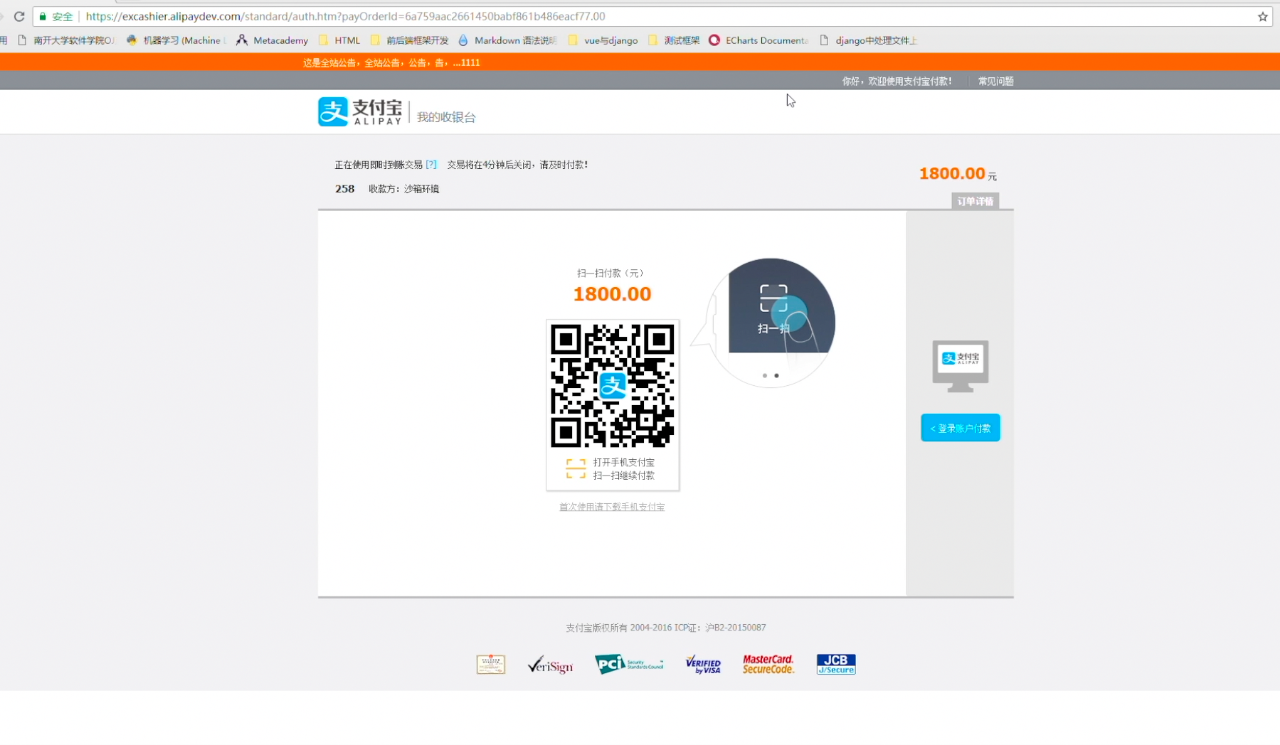
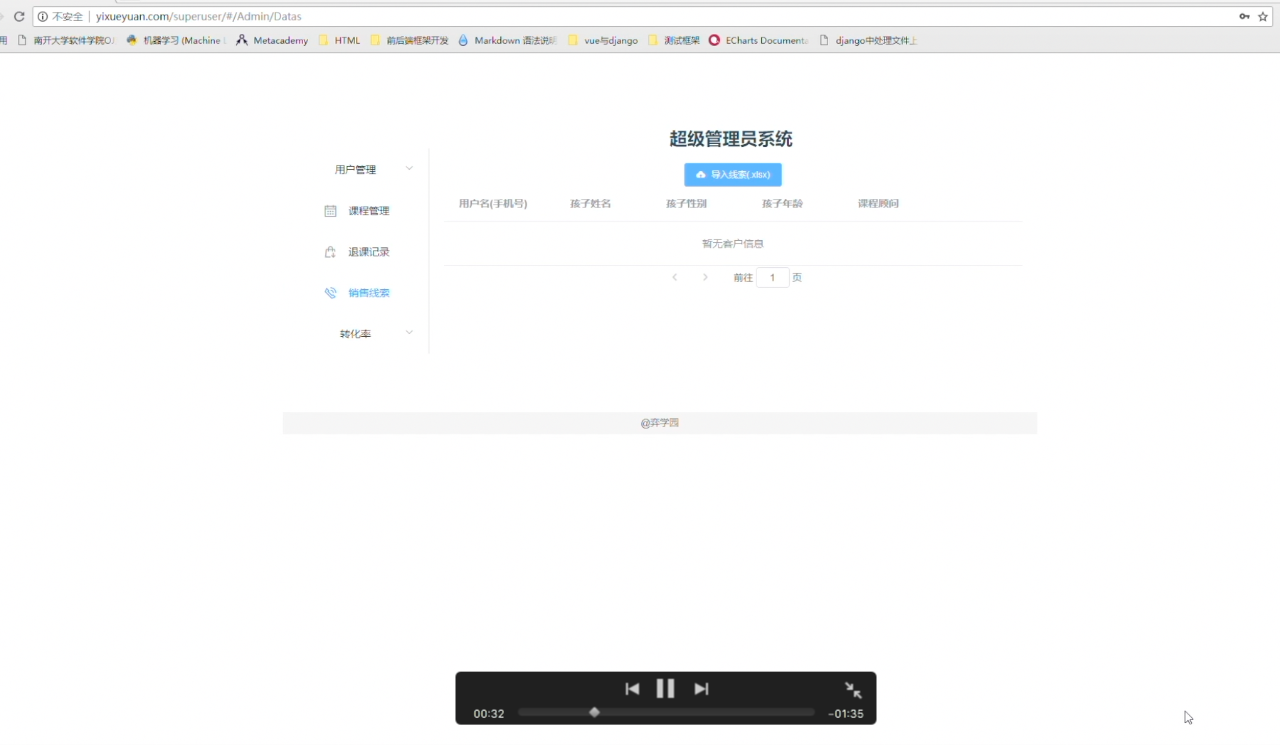
4.系统显示对应操作所需数据

5.超级管理员对所需数据进行更改

6.系统保存超级管理员操作结果

**三 项目效果**

**** ****

**** **** 

**学生对实训综合评价**

|  |  |
| --- | --- |
| **项目组名称** | **综合评价** |
| **魏鹏** | 通过本次实训，掌握了很多新的技术，但还有很大的进步空间，还需要自己多加练习，多去学习，争取把所学的技术熟练应用于项目中。 |
| **项目组名称** | **实训成果** |
| **魏鹏** | 1. 用户故事分析、撰写：用时3小时；在这个过程中遇到了团队成员意见不统一问题，解决；学会了如何团队沟通。 2. 脑图设计：用时3小时；遇到了脑图设计工具不熟悉问题，解决；学会如何使用百度脑图。 3. lint选择，git-hook配置:用时两天；遇到了文档搜索困难，英文文档无法理解，不会写脚本，安装后无法使用等各种问题，解决；为团队成功配置git-hook。 4. 项目流程了解，任务拆解分配；用时3小时；遇到了不知道该如何配置环境，不知道接下来要做什么，怎么做，解决；学会了团队合作，一个人学会再教其他四人。   5.设计编写数据库：用时7h  主要问题：  设计user表的时候，因为要利用Django自带的auth模块，所以需要  继承auth模块，不知道如何继承。  解决：  通过阅读官方文档，选择了使用OneToOneField字段继承auth模块中的user表。  收获：  了解了model模块中各种字段的使用。  6.实现支付宝支付：用时两天半  主要问题：  （1）支付宝SDK没有python版。  （2）支付成功后的异步回调通知无法到设置好的回调通知地址  解决：  通过阅读各种中英文文档，找到了轮子Django实现支付宝支付功能，  及暂时使用同步通知。  收获：  学会了利用Django如何为网站实现支付宝支付功能，及支付宝支付流程的深度理解。  7.编写Django单元测试：用时两天  主要问题：  不知道怎么写单元测试。  解决：  阅读官方文档。  收获：  学会如何写单元测试，提高了英语阅读水平。  工作内容：  8.测试应用的各方面使用：用时7h  主要问题：   1. 由于一直写后端api和测试不是很熟悉前端页面设计，导致测试缓慢。   解决：  通过队友的帮助完成应用测试。  收获：  找出了项目中的一些细节上的bug，提出并和队友一起改正，我们的项目变得更完整优秀。  9.完善支付系统：用时4h  主要问题：  （1）最后一周为了跨域使用了nginx，导致支付系统出现无法获得请求的问题  解决：  通过阅读文档和队友帮助，了解了nginx的使用，解决的此问题  收获：  学会了如何使用nginx，解决支付跨域请求问题。  10.撰写功能测试报告：用时10h  主要问题：  没有什么问题，在对应用进行测试之后，功能测试报告比较好写，主要问题就是需要按要求写的非常细致，需要花费很多时间。  收获：  再次测试出一些小bug,学会了如何写功能测试报告。  11．交叉测试，测试其他组的应用：用时15h  主要问题：不熟悉其他小组的环境搭建，而且其他组的部署文档写的并不是十分详细，环境搭建失败了很多次。  解决：通过询问要检查小组的同学和仔细阅读他们的部署文档、readme文件。  收获：学习到了其他小组项目设计和代码设计上的一些优点，也同时给他们的应用找到了一些代码风格和功能交互上的问题，为自己的小组得到一些分。  12．git问题：本周发现将本地git仓库文件提交到远端仓库后，当发起合并请求，请求中会多出一些很久之前的提交。  问题原因：是本地将master分支merge到工作分支，导致了这个问题  解决：在发起合并请求时勾选Squash commits when merge request is accepted.  收获：会在之后注意不能操作失误而将master分支合并到工作分支。 |