

# 北京邮电大学信息与通信工程学院

## 本科毕业设计（论文）中期进展情况检查表

学院	信息与通信工程学院	专业	信息工程	班级	2018211117
学生姓名	陈俪影	学号	2018210476	班内序号	14
指导教师姓名	朱旭振	所在单位	信息与通信工程学院	职称	副教授
设计（论文） 题目	（中文）基于 Django3 和 Vue3 的高招志愿填报辅助系统开发				
	（英文）The development of auxiliary system for filling application of college entrance examination based on Django3 and Vue3				
目前已完成任务	<p>一、任务目标</p> <p>（一）主要任务：</p> <p>任务 1：阅读相关文献，调研利用 Python 开发 Web 后端系统的方案，学习当前流行的 Web 前端技术，同时整理高招志愿填报辅助系统的业务需求。</p> <p>任务 2：基于 Django3 开发后端系统，处理服务逻辑，而基于 Vue3 开发前端系统，处理与用户的交互逻辑。此外，基于 MySQL 数据库存储系统数据。</p> <p>任务 3：基于 Nginx 实现系统部署，同时对系统功能和性能进行充分测试。</p> <p>（二）工作内容：</p> <p>内容一：阅读相关文献，调研基于 Python 的 Web 系统开发方案，分析高招志愿填报辅助系统的业务需求。</p> <p>支撑毕业要求指标点：3.1/3.3/5.1</p> <p>3.1:了解现有的基于 Python 的 Web 服务器开发技术现状，同时了解当前比较流行的 Web 前端开发技术。此外，对高招志愿填报辅助系统的业务需求进行梳理，同时考虑系统的管理功能和安全性。</p> <p>3.3: 基于所了解的基本开发技术，设计基于 Python 的高招志愿填报辅助系统技术方案。</p> <p>5.1: 在进行调研的过程中，掌握文献搜集检索的工具和方法，通过文献学习基于 Python 的后端 Web 开发技术以及流行的前端开发技术，作为系统设计参考。</p> <p>内容二：基于 Django3 开发后端系统，处理服务逻辑，而基于 Vue3 开发前端系统，处理与用户的交互逻辑。此外，基于 MySQL 数据库存储系统数据。</p> <p>支撑毕业要求指标点：4.1/4.2/6.2/10.1/10.2</p> <p>4.1:根据项目目标，首先，掌握 Django3 后台开发技术，明确 url 的跳转控制、视图业务的实现、模板文件的开发以及与 MySQL 数据库的绑定方法；其次，掌握 Vue3 开发前端系统的一般技术流程，明确状态管理 Vuex、前后端通信工具 Axios、国际化、Element Plus 控件等关键实现技术；最后，掌握 MySQL 数据库的表设计，明确业务对数据库设计的要求。</p> <p>4.2:综合上边的各个功能块，开发总体应用，不仅能实现新增院校或专业的信息进行添加修改删除，而且能实现考生注册、查询院校、对比学校、专业历年分数线、估分选择学校、估分选择专业等功能。</p>				

	<p>6.2:在搭建好高招志愿填报辅助系统框架后，与市场上已有类似软件进行对比，在对照分析中优化提升，理解、评价本系统及所涉及相关技术可能对社会、健康、安全、法律以及文化带来的影响；</p> <p>10.1:在调研、技术交流、汇报分享等过程中与同学、实验室学长、老师和社会上的技术专家进行有效的沟通交流；</p> <p>10.2:在文献阅读、技术学习以及分享交流等环节，充分借助外语学习外文资料、增强国际视野、跨文化沟通合作。</p> <p>内容三：基于 Nginx 实现系统部署，同时对系统功能和性能进行充分测试。</p> <p>支撑毕业要求指标点： 4.1/4.3/6.2/10.3/12.1</p> <p>4.1:明确基于 Nginx 的 Django3 部署方法，实现 Web 工程的部署；</p> <p>4.3:对系统的用户管理功能、院校或专业的信息管理功能、考生估分选择专业和学校等功能进行可用性和可靠性测试，需要保证同时 10 个用户申请访问时不会拒绝，同时所有功能正常工作。</p> <p>6.2:将本系统采用的实现方式与其它已有方式的进行比较分析，理解、评价本系统及所涉及相关技术可能对社会、健康、安全、法律以及文化带来的影响；</p> <p>10.3:就复杂工程技术问题与业界同行有效沟通交流、撰写报告和设计文稿、陈述发言；</p> <p>12.1:认识不断探索学习的必要性，具备自主学习终身学习的意识，不断学习适应发展。</p> <p>二、主要任务完成情况</p> <p>（一）充分调研了目前高考志愿填报辅助系统的业务需求以及相关的产品情况，了解同类产品的各自优劣势，为自己的产品建立了产品理念、功能需求的设计，并且设计了一范式的 UI。</p> <p>（二）充分学习了解使用 Python 后端开发 Web 程序的解决方案，并学习了解前后端分离式开发原则、Vue3 前端框架、Django3 后端框架，以及 MySQL 数据库的部署，并且能够使其互通连接、形成完整的程序架构。</p> <p>（三）实现了基于 Nginx 进行系统部署的功能，并且在开发的过程中对正常使用和常见错误进行了测试，保证了运行的稳定性。</p> <p>三、实际工作</p> <p>（一）整体设计</p> <p>1. 用户使用的功能方面：</p> <p>账户管理系统：用户能够进行账户注册与登录，保存一些个人信息。并且能绑定邮箱或手机号以进行密码找回或重置。</p> <p>院校信息浏览：用户可以浏览数据库中的高等院校信息，包括院校的简介、开设的专业、历年的录取情况等，并且应当可以使用一些分类标准进行筛选。</p> <p>录取分数线预测：用户可以取得某学校新的一年录取分数或名次的预测。</p> <p>录取可能性分析：平台根据录取分数线进行预测后，根据得到的结果以及用户自己的分数或名次，根据一定的算法去进行录取可能性的分析。</p> <p>模拟志愿填报：用户可以体验志愿填报的形式，并且平台会根据用户的实际情况进行院校的推荐。</p> <p>一分一段查询：为用户提供报考省份的一分一段表，以供用户更好地了解自己的高考成绩。</p> <p>其他高校信息：提供如双一流评级、全国学科评估结果、各平台的高校排名等信息，</p>
--	---

	<p>以供用户浏览。</p> <p>高考新闻政策：收录一些最新有关高考的新闻政策，以供用户浏览。</p> <p>2. 后台管理的功能方面：</p> <p>用户信息管理：管理员能够直接获取用户列表，并增加、删除用户或修改用户的信息。</p> <p>文章发布管理：管理员能够获取文章列表，并新增、删除或编辑文章。内置富文本编辑器。</p> <p>院校库管理：管理员能够新增、删除或修改院校的资料。</p> <p>3. 程序设计方面：</p> <p>使用 Django3 开发后端、Vue3 开发前端，进行前后端分离式开发。</p> <p>Django 后端暴露接口，前端页面可以对接口使用 Axios 进行网络请求，获取需要的信息。</p> <p>使用 MySQL 数据库管理数据，并与 Django 后端建立连接，能够使用 Python 脚本进行增删查改。</p> <p>前后端分别建立测试服务器，使用 Nginx 反向代理到同一域名下完成整体部署，使得用户能够正常访问前端部分的页面，并解决 API 的跨域问题。</p> <p>（二）前期工作</p> <p>通过网络搜索的形式调研市场需求以及同类型产品的现有成果，分析学习其他产品的优势，并考虑如何创新、增加自己的内容。</p> <p>初步学习 Vue 前端框架和 Django 后端框架的相关知识，了解工程的结构以及前后端分离式开发的原理，并且了解前后端数据传递的机制。</p> <p>学习 MySQL 数据库的基础知识，熟练掌握数据库管理的基本 SQL 操作，并学习如何使用 PyMySQL 包来在 Python 脚本中访问数据库。</p> <p>通过学术论文网站等途径对目前常用的高考录取分数线或名次预测算法进行对比，初步确定自己的产品上使用何种预测算法。</p> <p>设计产品的功能需求以及站点结构，并考虑各模块需要收集什么样的数据，以及如何设计接口等。</p> <p>学习网站 UI 设计的方法原则，使用 Mac 平台的 Sketch 软件设计 UI，采用统一的配色方案，合理规划视觉安全区和区块比例等。</p> <p>设计数据库的结构，以备后续录入数据。</p> <p>（三）正式开发</p> <p>1. 数据收集</p> <p>从产品的功能方面出发，需要采集一些数据用以展示和分析，其中包括中国大陆高等院校的名单（大学的名称、所在地、序列号、办学性质等），还有各高校的简介以及往年的录取分数等。这些数据都尽可能从官方渠道获得，如果没有，再考虑从其他的正规网站上获取。</p> <p>主要采用的方法：①直接获取；②网络爬虫。</p> <p>2. 数据处理</p> <p>获取到的数据一般格式比较不确定，且经常混杂有一些其他的内容，需要采取一定的方法进行清洗，并且整理格式录入数据库。</p>
--	---

### 3. 算法设计

分析对比了不同的预测方法后，目前暂时选定的方法是使用线性回归算法进行名次预测。这里设计了一段脚本，可以根据学校、报考地区和选考科目，从数据库中获取近几年的数据转换成名次，并进行预测。

进行线性回归的方法使用的是 `sklearn` 包的 `linear_model` 类中的 `LinearRegression` 函数。

### 4. 搭建后端工程并建立模型

在 `PyCharm IDE` 上开发，建立了一个基于 `Django3.10` 的后端工程，并调试了相应的设置。其中，将各个模块在工程中分成了不同的应用。后端部分分为以下几个模块：

**main:** 提供分数线预测和院校推荐的后端支持，是应用的核心功能。其中在预测的功能里，预测的结果会自动保存到模型的数据库中，当再次预测相同的参数时可以直接提取库里的结果。

**users:** 提供用户账户的支持。其中账户验证系统使用 `REST framework` 的 `Token`，可以为每个用户模型生成唯一密钥来进行验证。这个模块中还拓展了 `Django` 本来的用户模型，允许增加更多的用户资料（高考信息如报考省份、分数等）。

**articles:** 建立文章模型，包括标题、发布时间、内容和分类。一分一段表、高校信息、新闻政策这几个部分都会使用文章模型来呈现。

**schools:** 提供院校资料库的查找支持。这里计划只用来提供从数据库查找学校的 `API`，无需创建模型。

### 5. 配置 Django Admin

`Django` 后端框架是配备有管理员账号系统以及后台管理页面的。在本项目中，网站管理员的后台管理也将使用 `Django Admin` 进行。

这里后台能够使用的的功能是查看用户账户、更改用户用户信息、查看文章列表、发布或编辑文章。

其中，使用 `TinyMCE` 插件给发布文章页面加入了富文本编辑器，以便管理员发布结构更丰富的文章。未来计划添加院校资料库管理的功能。

### 6. 建立前端工程并配置路由

前端工程的建立使用了 `Vue3` 目前推荐的 `Vite` 脚手架，开发过程也使用其推荐的 `Visual Studio Code` 加上 `Volar` 插件（`Vue3` 语法支持）的方式。

在开发过程中了解到，前后端分离式开发通常舍弃后端路由而采取前端路由的方式，后端仅提供接口。通常在 `Vue3` 的开发项目中，会采取单页应用的形式，使用 `vue-router` 分配路由并接管浏览器的导航操作。


完成了所有页面的路由建立以及首页重定向、`404 not found` 重定向等。

### 7. 将前后端的测试服务器使用 Nginx 反向代理到同域下

为了一劳永逸地解决跨站问题，把前后端的开发服务器使用 `Nginx` 放在同个域名下，使用 `url` 模式进行分离。其中 `admin/` 和 `api/` 前缀的属于 `django` 服务器，其他的则属于 `vue` 的服务器。这里的效果是直接访问网站会进入客户端首页，而访问 `admin/` 则会进入后台管理。

### 8. 前后端并行开发

	<p>从此开始就采用的是前后端并行开发的方法了，具体来说就是先按照先前的 UI 设计以及功能预期去开发前端页面，在需要请求接口的时候再针对性地设计接口，并直接使用 Axios 对接，然后测试。</p> <p>四、页面功能成果</p> <p>（一）首页：提供站点全部模块的链接导航。</p> <p>（二）用户模块：</p> <p>1. 注册/登录：基本的注册登录功能，注册时初步检测用户名和密码的合法性，并收集用户高考信息。</p> <p>2. 我的资料：提供查看和更改个人资料信息的功能。</p> <p>3. 安全管理：提供绑定手机/邮箱和重置密码的功能。（只完成了重置密码功能）</p> <p>（三）填报指导：</p> <p>1. 录取可能性分析：本毕业设计的核心功能之一，能够预测某学校的投档线，并结合用户的成绩或名次进行录取可能性的分析。</p>	
	是否符合任务书要求进度：是	
尚 需 完 成 的 任 务	<p>1. 完成院校资料库模块</p> <p>目前的数据收集已基本完成，只需要开放访问数据的接口，并完成前端页面的展示即可。</p> <p>2. 完成模拟志愿填报模块</p> <p>目前基本确定形式（分批次，依据用户的报考地区提供 n 个大学各 n 个专业的志愿选取模拟）。但仍需确定计算的方式，然后开放访问数据的接口，并完成前端页面的展示。</p> <p>3. 完成文章模块</p> <p>发布文章后后端模型会存储数据（文章内容为 HTML 形式）。需开放访问数据的接口，完成前端页面。</p> <p>4. 补充一些文章数据</p> <p>5. 在后台管理中加入修改院校资料库的入口</p> <p>因为院校资料库目前不是使用 Django 模型创建的，而是直接被写入了 MySQL 数据库，所以暂时还没有注册到后台管理。</p> <p>6. 考虑更好的预测算法</p> <p>目前采用的线性回归算法相对比较简单，不一定能够比较好地对今年分数进行预测。考虑其他的预测算法并进行选择。</p> <p>7. 使应用适配更多的报考省份及分科模式</p> <p>目前应用仅收集了部分文理分科的省份的数据，并只对文理分科的模式进行了适配。后续将尝试适配更多的报考省份及分科模式。</p>	
	能否按期完成设计（论文）：是	
存 在 问 题 和 解	存 在 问 题	<p>1. 模拟志愿填报模块</p> <p>需要考虑存储空间、计算时间和准确程度的优先级问题。如果采用全部提前预测并存储在网站服务器上的方法，会占用大量存储空间；如果采用实时预测的方法，既需要一定的存储空间，又会产生大量的运算时间；采用单次分数或排名的方法可能存储量小且比较快，但准确率不如预测后高。需要进行一个选择。</p> <p>2. 在后台管理中加入修改院校资料库的入口</p>

决 办 法		<p>因为现在使用的数据表不是 Django 模型,所以加入后台管理会需要额外的步骤。</p> <p>3. 更好的预测算法 录取分数/排名线预测涉及的数据呈现数据量少, 且有一定的不确定性的特点, 需要视情况选择好的算法。</p> <p>4. 适配更多的报考省份及分科模式 其他的分科模式需要考虑学校的录取限制, 而且需要改变目前的数据库结构(目前的存储结构是文科分数和理科分数)。</p>		
	拟 采 取 的 办 法	<p>1. 模拟志愿填报模块 考虑到名次较分数的不确定性更小, 计划采用最新年份的录取名次进行推荐, 牺牲部分的准确度, 从而更快地获取结果, 并无需大量的存储空间。</p> <p>2. 在后台管理中加入修改院校资料库的入口 Django 内置的工具中有将数据表转换成模型的工具。计划使用该工具将原本的数据表转成 Django 模型, 然后注册到后台。</p> <p>3. 更好的预测算法 目前会考虑将现有算法与 ARMA 模型(自回归滑动平均模型)等方法进行准确度的对比, 并持续调研文献资料等, 寻找更好的预测算法。</p> <p>4. 适配更多的报考省份及分科模式 按照原先的方法继续收集数据, 并且调整存储数据结构使其兼容不同的分科模式。</p>		
指导教师签字			日期	2022 年 4 月 12 日
中 期 检 查 小 组 评 分	评价内容 (BY: 毕业 要求)	具体要求		评分
	选题与调研 (BY5, 6, 12)	符合专业培养目标, 能够支撑毕业要求, 考虑了社会、健康、安全、法律以及文化等非技术因素, 难易适度; 能够根据任务书要求独立开展文献查阅和相关调研, 对调研获取的信息能够进行归纳总结, 具有学习新知识的能力。(满分 7 分)		6
	设计(论文) 水平与进度 (BY3,4)	能够按照任务书规定的要求提出符合选题的可行性研究方案, 独立进行实验(如安装、调试、操作)和研究方案论证, 能正确处理实验数, 能对课题进行理论分析和研究, 得出合理有效的结论, 工作进度符合预期。(满分 8 分)		6
	内容表述与问 答 (BY10)	语言表达清楚准确, 思路清晰, 逻辑性强, 论点正确; 回答问题准确、基本概念清晰; 中期报告撰写基本规范, 图表完备。(满分 5 分)		5

	中期检查评分合计（满分 20 分）：17
	评语：毕设进度满足中期阶段的要求，进展良好，建议后期完善独创性部分。
	中期检查小组组长签字：检查日期：

	<p>中期检查评分合计（满分 20 分）：17</p> <p>评语：毕设进度满足中期阶段的要求，进展良好，建议后期完善独创性部分。</p> <p>中期检查小组组长签字：_____ 检查日期：_____</p>
--	--

	<p>中期检查评分合计（满分 20 分）：17</p> <p>评语：毕设进度满足中期阶段的要求，进展良好，建议后期完善独创性部分。</p> <p>中期检查小组组长签字：_____ 检查日期：_____</p>
--	--

	<p>中期检查评分合计（满分 20 分）：17</p> <p>评语：毕设进度满足中期阶段的要求，进展良好，建议后期完善独创性部分。</p> <p>中期检查小组组长签字：_____ 检查日期：_____</p>
--	--

	<p>中期检查评分合计（满分 20 分）：17</p> <p>评语：毕设进度满足中期阶段的要求，进展良好，建议后期完善独创性部分。</p> <p>中期检查小组组长签字：_____ 检查日期：_____</p>
--	--