**个人总结**

519021910162 鲁为涛

1. 项目初衷与初期设想

我们小组成员来自3个不同专业，3个不同年级，因此有比较多元化的专业背景和个人兴趣爱好，在讨论时也能基于不同的专业知识互相提供宝贵的经验。这也是我第一尝试和不同专业的同学们合作完成一个软件开发。我的经验可能对后续选修该课程作为交叉课程的同学也有一些启发, 希望你们认真对待本课程，因为可以学到很多知识。

在最初立项讨论时一共有两个方案，分别是由我提出的实验室管理软件和黄悦珊同学提出的校园流浪猫狗档案管理平台。

我提出实验室管理软件的初衷是因为大三同学在面临选择实验室时，时常发现难以获取到实验室的消息，而我的导师在发布实验室招募信息时，也难以沟通到有需求的同学，因此我想要设计一个软件帮助师生间更好的打破消息壁垒。黄同学提出的校园流浪猫狗档案管理平台首先是因为她本身是社团成员，平时该社团内本身缺乏系统性整理，社团内的救助信息也难以及时被关注，社团外的同学更是缺乏与救助人员的联系渠道和对校园猫狗的姓名和喂食、生活习惯等信息的了解。

经过讨论，这两个设想都有很不错的实现价值和用户需求，但考虑到实验室网站可能仅需介绍的首页，和网上很多开源的“图书管理系统”功能实在过于类似，在技术上可能难度较小，此外很多教授也可能不喜欢统一的宣传设计风格，或是和其他实验室放在一起做比较，真实的实用性可能不高，而校园流浪猫狗档案管理平台则没有这些顾虑，首先它包含了上述实验室网站的展示信息功能，其次也可以增加更多的有趣功能，锻炼到更多的技术，最主要的是喵汪社团目前没有该功能，我本人在实际生活中也发现校园内的猫猫狗狗确实很受同学们的喜爱，但经常被“错爱”，例如投喂不能食用的咸火腿等，因此在听闻了黄同学的项目设想后我也较认同开发该项目。

最后我们的初期设想是制作一个校园猫咪流浪狗档案管理平台，记录猫咪/狗狗的基本情况、检索、科普征名、领养寄养信息发布。

最终经过一个学期的开发，我们小组成员最终实现了我们的这些初期设想，我实现了喵汪在线交流平台的功能，于我个人而言，我学习了许多崭新的技术和软件工程的开发思想，从零开始逐渐精通了一些主流框架，尝试了软件小组开发的过程也认识到了许多的不足，吸收了很多经验教训，感觉收获匪浅。

1. 个人总结

2.1 项目中git的使用

我们小组首先学习了往届的优秀经验，参考了陈子璇学姐提供的经验（如图1所示），为了避免同一个功能多人开发，导致沟通、代码版本冲突的问题，或者分工后合作的同学进度较慢导致无法进一步开发的情况，我们小组采用的是一个功能一个分支的工作模式。

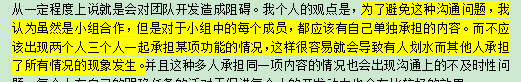


图1 参考的优秀文档

整体分工我们小组一共有3个分支，分别是许慎林同学负责的档案管理后端分支和秦培翔同学负责的档案管理前端分支，以及我负责的在线交流平台分支。 主分支中是一开始学习git、学习框架时的文件。

我自学的git工具是git GUI，后来在第二次组会中许慎林同学介绍了使用vs code实现git的方法，因为我们小组的同学本身就比较熟悉vscode的使用，比较后觉得使用vscode pull和push都非常的方便，因此基本上放弃了git GUI的使用，后续我们小组的git开发模式就是git clone后使用vscode更新。

在开发后期，我将在线交流平台部署到了阿里云服务器后，采用了热更新的指令实时构建前端页面（npm run serve）使用该指令后对css、vue文件的修改会直接在浏览器中显示，非常方便。此后我将用vscode登录阿里云服务器的方法教给了黄悦珊同学，她可以直接看到每次更新后的前端样式并给我提供宝贵的设计思路，并且帮助我修改前端样式。与服务器的文件传输使用的是filezila软件。

一人一个分支的优点在于我们有效规避了团队开发中进度不统一、多人开发同一功能导致有同学划水摸鱼的情况，我们每一个开发的同学都非常清楚自己的责任并且可以按自己舒服的时间安排和节奏执行开发，开发效率非常高（很早就实现了前后端功能），也因为一人一个分支，每个人都可以尝试自己喜欢的技术，所以我们最后项目相比其他项目整体上实现了较多不同的功能，也用到了许多不同的技术，缺点就是在打磨整合阶段遇到了许多问题，比如许秦同学之间前后端的跨域问题、开发时的需求沟通（他们是比较熟的朋友，直接用手机沟通了，但后期我就不太清楚需求的新变动），可能导致了我们最后的功能有一些割裂感。

在整体学习阶段和后期整合阶段我们的git更新比较多，开发阶段大多数因为每个人功能独立，大部分都在本地直接修改了导致git不足。

**/\*虽然根据报告要求应该区分写个人提升和工作**，但我觉得的我的提升是伴随着项目进行过程不断增加，所以我先分别做简要总结，再按时间顺序详述整个开发过程中比较重要的这些提升和工作。（合并写，正文部分总计5页，不超过篇幅限制8页）\*/

2.2 个人技能提升的总结：

我在本课程中提升的技能:学习了Git能力，对vscode使用能力提升(git方法、python虚拟环境、远程资源管理（ssh远程服务器）、云服务器环境配置等)。后端：对Django，mysql有了解，从零开始熟练使用node.js，学习了javascript。前端：熟练掌握vue3框架，熟练使用css、html网页设计，实现了完整的由自己设计的前端css，部署：熟练掌握华为云的自动构建编译+部署、探索了阿里云部署方法。学习了项目需求、团队管理的软工知识，学习了华为云的项目管理工作。理论：对websocket协议有了更深入的了解。

2.3 个人工作的总结：

开发：我主要负责在线交流平台功能的开发，先开发了能通讯的终端，再根据需求改造功能最终实现在线交流平台。完成了在线交流平台的后端开发、前端开发（与黄悦珊同学合作）、在线部署，补充团队需要的学习资料、督促团队进度，在学习方面参考网上教程学习了图书管理系统1.0和2.0两个Django项目，学习了大量bilibili的教程、csdn博客和github代码仓库。

2.4 个人整个课程中的工作和学习经验

我来自信息工程专业，目前大三，选修该课程的主要原因是因为我系统地学习了较多的编程、算法理论、通信协议等知识，但却没有过将其应用于实际开发的经验。而我本人其实一直对软件开发有一些兴趣，将来也希望在该方向有更深入的研究。可惜以前专业课比较多一直没有时间，这学期课较少，因此选修了本课程，希望了解真实软件开发的过程，学到更多有趣的技术，挑战自我。这学期在这个在线聊天平台上我投入了非常多的时间，学习了大量前后端知识并应用在项目中。

我的学习经验和工作按时间顺序可以分为以下几个部分：

**准备阶段**（熟悉课程和小组同学，Git学习，框架学习）：

**熟悉同学**：

我们小组是临时组队的，来自不同专业，同学大多彼此不熟悉，因此我首先在群里询问了每一位同学的技术基础，我们小组成员的初始能力如图2所示：

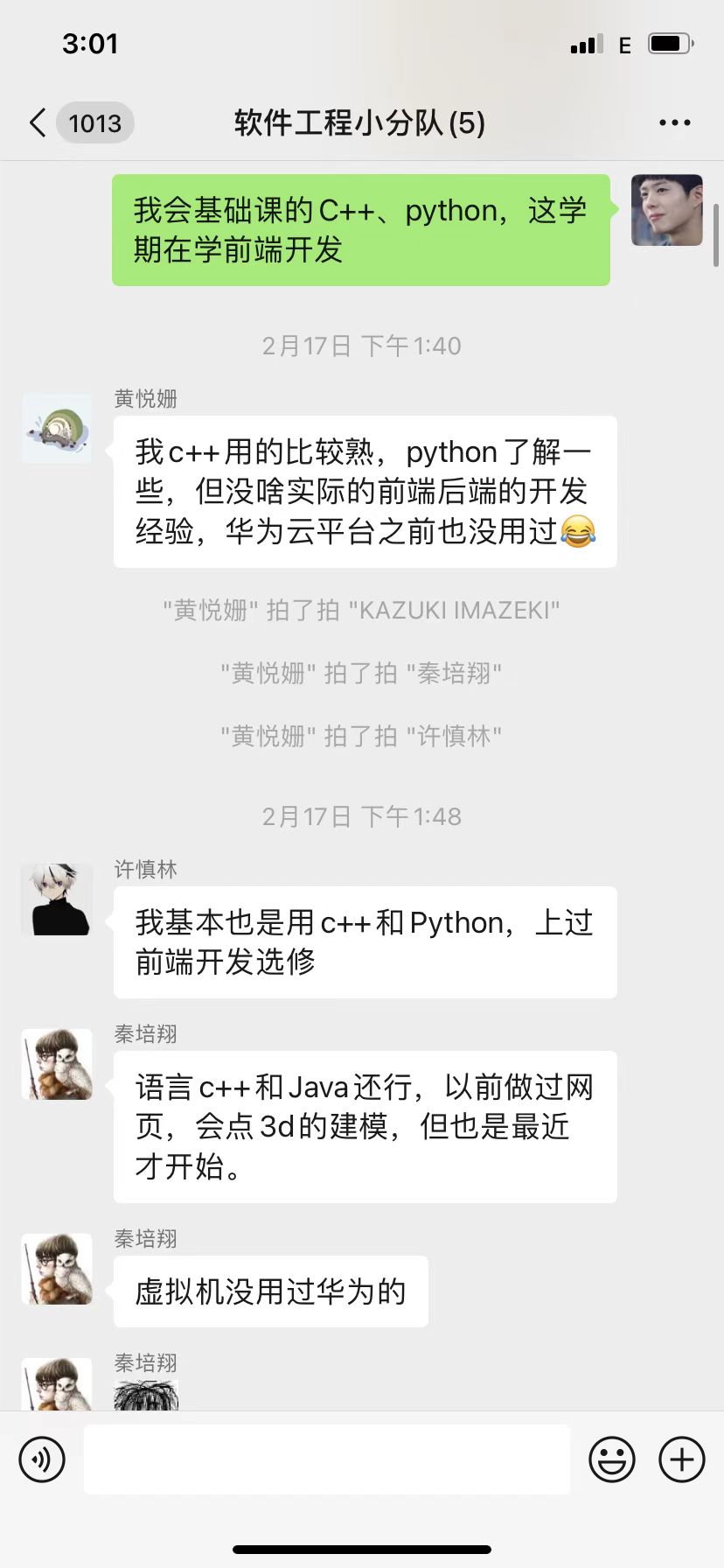


图2 交流基础

我们小组的基础能力是相当不错的，两位信安的同学都有过网页开发的经验，并且我们都有较好的编程基础，大家也因此比较有信心，经自荐，许慎林同学成为了我们小组的组长。

在第一次和第二次组会中，除了正常的讨论项目立项、git学习外，我们也交流了本学期的课余时间、目前上的一些专业课等话题，这些“唠嗑”加深了我们成员间对彼此的了解，也让我们分配任务时大概了解了每个人可以接受多少任务量。

**GIT学习**：

首先我个人学习了廖雪峰的git教程，熟悉了华为云的仓库使用和用git GUI的git方法，此处与课程基本要求一致，不多加赘述。值得一提的是在第二次组会中，许慎林同学向我们演示了gitclone代码分支到本地后，通过vscode直接git pull和commit&push的方法，对比git GUI非常高效，因此此后我们小组在开发中都采用了vscode git pull&push的方法。



图3 使用vscode轻松实现git push&pull

**框架学习**：

整体上而言，我和许慎林同学在进度上是较快的，在框架学习的任务中，有开发经验的许慎林同学直接尝试了完成宠物档案管理的基本功能，而我则是去网上学习了一个图书管理项目的Django教学视频，并在网上找到了对应的代码和一些Django的教程。我按照视频一步步学习，途中却遇到了许多视频中没有出现的bug，我便复制错误信息上网查询，大多是数据库的配置文件中的初级错误和网页路由错误，如“未创建表”等，至于python环境则是按照教程推荐的版本用Anaconda建立了Django专属虚拟环境，因此在开发中并没有遇到往届经验中常见的版本不一致问题。（该问题在后续nodejs开发时大量出现了）。此后我们便进行了第三次组会讨论学习任务和后续分工，我在会上分享了Django项目的学习成果并作为作业提交，许同学制定了较为合理的后续学习任务，我则在源码上增加了注释、删除了多余功能，并分享学习文件供其他同学学习，如图3所示。经过本次讨论，我们小组基本确定了分工，我和许同学负责后端功能的实现，秦、黄、今关同学负责前端部分。该Django项目中的路由和数据库使用方法让我对Django有了较全面的了解。



图4 学习任务制定和框架学习

https://www.bilibili.com/video/BV1su411Z72Y?spm\_id\_from=333.337.search-card.all.click

**开发阶段:**(兴趣引领，需求适应，遇到bug不要慌)

对于第一次进行开发的我，整个开发阶段就是和数不清的bug战斗，不断学习的过程。由于许慎林同学很早便基本上完成了猫咪展示的后端设计，秦同学则在负责该功能的前端设计，于是我就把开发的重心放在了与用户交互的功能，例如发布救助帖、上传猫咪信息等需求上。该需求有许多的实现方式，一开始我们考虑的也是采用和后端上传类似的实现方法，但从本质上只是把数据库的输入放在前端实现，我感觉这样的技术实现实在有点简单。。和我选课时希望的技术提升有所不同，所以我经过思考，采用类似淘宝客服的在线交流平台的设计思想，既能够保证原有的发布救助，上传猫咪等需求，也能够为急需的用户提供实时帮助。

我之前在专业课中学习过应用层协议，实现过类似tcp协议的终端间通讯，在前端课程中了解到了目前较前沿的前端websocket协议，正好自己此前一直有一个写聊天室和好朋友“私聊”的小心愿，于是也想自己尝试一下，将专业知识融入进项目里。我首先按学习任务完成了vue框架的学习，又摸索了一个周末，先系统学习了websocket官方的文档和demo代码，学习了一些websocket库api调用的博客，终于参考github的代码用js实现了类似之前tcp协议的本地多终端间的websocket通讯，这也是后续整个交流平台的雏形。我将该代码上传至了自己的分支。我首先想使用的是集成好的socket.io库，但后来部署时遇到的bug太多，并且我们的功能还是比较简单的，所以改用原生websocket内置的函数，刚开始使用js还是比较难上手的，经常还带着C++编写时候的习惯，变量类型也傻傻分不清，但好在网上websocket相关的代码和服务器写法还是比较多的，遇到bug就查，最后倒也写出来了（服务器端代码量还是比较少的）。

在第一轮迭代的时间点，我参考github的代码框架用vue3的框架实现了基本的客户端页面，但仅仅是单房间的多人聊天，没有任何前端设计，和常见的社交软件实在相去甚远。所以我希望实现更多满足用户需求的功能。

由于是基于通信协议，后端服务器只是起收发包的作用，简单的修改后端服务器的代码是无法在前端实现这些功能的，但是我们小组内和我对接的前端同学的学习进度此时还比较少，于是我又开始学习前端知识（同时我也在上前端课程，所以对基本的html，css还是比较熟的），在网页设计方面，我没有学习多个vue的路由跳转（学Django的时候感觉bug好多），而是利用vue的特性，在一个vue文件中使用v-if语句通过变量实现局部渲染，从而展示不同的页面（详见源码），避免了路由的问题，并且理论上这种方法可以减少重复渲染相同元素（但我没优化），在功能方面，我学习了大量网上的源码和博客，比较实现和迁徙的难易，将核心代码找到并迁徙到自己的项目中，例如我们项目中的随机取名、自动生成头像功能、交流页面时左右侧的聊天头像显示、右侧的消息滑动条等功能，这些小功能的实现都是一个个从各种github源码中找到并迁徙过来拼接而成的。以随机取名为例，本身只是一个生成随机姓名的js文件，我将其引入到vue框架中，并参照vue框架需求的格式将js文件的路由配置正确，在.vue文件中的method中引用js中export出的类并实现了随机姓名函数的编写，再在主页的html文件中用合适的div渲染。在上述的每一步基本上都会碰到bug，解决bug的方法基本就是查询网络博客，但这些博客上的方法很多都是不同框架，不同版本Vue（大部分是Vue2的解决思路）中遇到的，使用nodejs开发虽然功能强大，但库更新实在太快了，经常遇到package版本不一致的问题，此外很多代码是用jquery代码写的，在迁徙的过程中我基本上举步维艰，只能采用遍历的方法尝试所有博客的解决思路，终于一步步把bug解决好了。

不过debug本身是一个越来越熟练的过程，经过大量的debug，后期开发中我遇到js相关的bug时基本在查询博客后看一部分就可以知道是否对应我的bug，以及是否是因为版本不一致导致的问题了，找功能也是类似，一开始可能要看完整个教程一步步做很慢，对代码熟悉后点开教程先看实现效果，再看它的源码，就可以直接找到自己需要的核心功能并迁徙到项目中。

在websocket服务器的编写中一开始凭自己的感觉写经常遇到bug，后面发现网上的源码和博客中的函数和框架都大同小异，索性按着网上博客中代码的框架写服务器，遇到的bug就少多了。此外，网页console也是很好的调试工具，在涉及端口请求、通信协议、渲染内容等问题方面，将console的报错信息在网上查询可以很好的解决问题。整个开发过程我花费时间最多的就是debug，但该过程也让我受益匪浅（尤其对整个vue3框架及其和vue2框架的区别，对vue框架常用库的版本兼容性问题、对websocket的原理和收发包时可能的bug有了彻底的了解）。

**部署阶段：**(官方文档，官方文档，官方文档！)

5月初，经历了开发阶段的bug战斗，我终于来到了部署的阶段，我本以为自己对整个代码和框架都理解透彻，已经所向披靡了，却没想到这里又是一场残酷的bug大战。

首先，我发现，我们小组初期的分工（前后端分离，每个人实现不同功能），导致此刻大家的进度不太统一，部署必然存在先后。最后我们小组的项目一共部署在了三台服务器上，这导致了猫狗展示页面前后端的跨域问题（许慎林同学在展示时已经有部分介绍）。

于我而言，vue3框架是最近几年最火的框架，在网上有非常多的部署博客和视频，我首先参考了一个视频，使用华为云自带的工具将其部署到了华为云上，但一直报错websocket connect error。经过思考，是因为我的项目中有一个需要用终端运行的服务器，但是网络上部署的仅仅是客户端，无法建立与服务器的连接。华为云自带的部署功能（构建部署以及上传打包文件等）都无法解决该问题，并且我没有注册华为云学生账号，调试的时候很快便欠费了。重新思考后，我采用了阿里云服务器，在该服务器上的环境配置、部署等问题花费了我大量的时间，我参考了许多网络视频中的“宝塔面板”来部署，并且熟练掌握了在vue3和vue2框架中修改ip来适应部署的方法（vue3中打包文件时省略了vue2框架中的vue.config.js，因此我手动补充了一个并修改打包时的publicpath），源码中的websocket请求也做了相应修改。但即便如此，宝塔面板也只适合部署打包好的dist文件，并不能解决服务器js无法运行的问题（需要在终端运行node server.js），怎么办呢？

最终我查看阿里云的官方文档，从它的例程中获得了启发，成功实现了部署。在所有项目基本开发和部署完成后，我和黄悦珊同学一起修改了样式、内容，又新增了管理员功能，修改网页样式和许秦同学的开发结合起来，具体部署细节限于篇幅，我将在工作特点章中进一步展开。我以前一直喜欢查博客和github，但这次甜头才让我发现，官方文档往往是最直接有效的解决方案！

**展示阶段：**(开发的同学也要能充分表达产品需求)

在小组展示之前，我们小组专门开了会讨论展示的内容，由于此前开发的整合只是在微信中私聊联系，没有很深入的交流，因此在会上主要是我和许同学互相交流，通过交流，我对许慎林同学的技术有了比较深入的了解，也对接下来的展示做了很详细的技术问题方面的准备（包括在展示中呈现的设计和优化细节）。 但没想到老师的提问主要针对的是我们项目的功能是为了满足什么需求设计的。当问到为什么用瀑布流展示猫咪信息，似乎喵汪公众号也有类似的功能？我一时语塞，因为不清楚喵汪是否有该功能，心里也在想是不是许同学也是和我一样，想尝试一下新技术才这样的？提问为什么要用在线平台而不用发布求助帖时，我也没有很好的回答，虽然我知道自己的设计初衷是尝试学到的websocket协议，也是为了给用户提供及时的求助渠道（我自己曾在喵汪群中发布过救助消息，确实很少有回复），但老师和听众们可能并没有类似的经历，所以会觉得有点多此一举。

展示后与比较了解我们项目需求的黄悦珊同学交流后（本项目的立项提案与大部分的文字内容都是她梳理的），首先发现了喵汪并不具备我们开发的功能，第二是目前救助队的消息回复并不及时，在线交流平台的设计可以让更多的同学都参与到救助者的身份中，也可以让救助队同学更好的处理救助消息（喵汪微信群内消息经常被其他消息刷掉）。

比较遗憾的是我们没有提前对需求和功能的统一做准备，在展示时也没有主动请黄同学来讲解需求，导致留下需求分析不充分的印象。在整个开发的过程中，我们小组确实一开始对需求有比较明确的认识，一开始的需求文档也写的很详细，但不断尝试新技术后，其实反而有一种“画蛇添足”的感觉。在真实的开发中，要时刻记住用最合适的技术实现需求，而不是用最fancy的技术。

在软件开发这门课中，尝试一下新技术是没问题的，但是需要在展示的时候说清新技术的优势，和它的优势为什么可以响应你们的需求。（我们小组展示时的问题就是讲清了优势，讲不清和需求的响应。）

在往年的个人总结中很少看到有关展示的思考，所以我在这里分享我们小组失误的经验，供后续同学们在开发过程中和展示前参考，不能只钻技术的细节，软件开发中要时时刻刻记住需求!

有同学在展示过程中恶意dos攻击其他小组的项目，有人在我展示的时候在我的交流平台中发布一些不恰当的粗俗言语，尝试了一些简单的dos攻击（比如打开很多页面进频道发送“攻击测试”的聊天记录。。服务器是能看到的。。还好我们页面渲染时候用了懒加载，所以这是可以的），希望以后的同学不要有类似的行为，大家用的都是最便宜的服务器，带宽本来就不够。

2.5 小组团队合作的总结

在本次项目开发过程中，我们小组内的氛围比较融洽，整体的合作也比较愉快。一开始的例会采取的是线下例会的方式，每周一次，第一次组会大家互相认识，讨论了项目需求，一起学习优秀总结，大致确定了项目idea和用到的技术栈（因为大家都有python基础，决定使用Django），后来一起线下学习git的使用，许慎林同学耐心的教会了我和黄悦珊同学，我们再去教今关叶月同学，大家都掌握了git的使用。

可惜天有不测风云，疫情的来袭让线下例会成了泡影，到线上组会后效率明显降低了。一开始每周的例会主要就是汇报自己的进度和学习的内容，我在网上学习了一个很不错的Django教学视频，跟着视频一步步实现了一个图书管理系统，在例会上和组员们分享并让大家一起学习，许慎林同学则分享了自己的工作进度（非常快，在框架学习时就完成了不少喵汪信息管理的功能）。例会后我会将文档和源码分享在微信群中，并且我和许同学完成了大致的任务分工，秦黄今同学主要负责前端知识的学习，我们俩主要做后端，接下来的任务是没学Django的同学过一遍我发的代码，学习一下Django框架和数据库的使用，大家也同意了。

但接下来可能因为课业比较繁重且疫情影响较大，我们小组的进度一度停摆了一段时间，组会也多以线上交流的形式进行，原定的例会是每周交流开发进度，但实际上例会经常有人缺席或以私聊的方式解决了需求，作业基本是辛苦许同学一个人按照一开始讨论的需求分析完成了。前后端开发的许秦同学主要用qq实时沟通需求，4/24完成第一轮迭代时（此时秦同学已经基本完成了所有猫狗信息管理的前端设计，许同学也完成了较多功能），我已经完成了网上图书管理系统2.0版本的教程,对Django更加熟悉，并且开始尝试websocket技术，实现了简单的聊天室。（但之前说好去学习的同学没有任何反馈，也没有说自己有新的进度）

在5月初的时候，我在微信群里说了一下自己的进度和想用聊天室实现需求的想法，但是没有同学回应。。后来问了黄同学，她也没有什么进展，可能大家那段时间都觉得第一轮迭代已经开发的差不多了，所以有些松懈了。此时我把后端基本完成了，实现了前端初步的页面，和许同学交流了部署的需求，获取到了公网ip。考虑到大家可能比较忙，而我正好这学期时间比较充裕，我就自己探索了一下如何部署在华为云、阿里云、如何设计前端页面，最后完成了在线交流平台大部分的部署和设计工作。当五月中旬许秦同学的代码上传到华为云仓库后，我查看了他们的代码，参考他们使用的CSS样式、图例，将自己的在线交流平台重新设计了一番，采用了相同的图标、类似的渐变背景等等，并和许同学一起实现了整合（网页跳转和功能按钮）。

整个开发过程中，我们三个主要负责开发的同学就在自己的分支上独立开发，由于两个信安的同学都有开发网页的经验，我这学期也学习了前端知识，独立开发了一些小网页，所以我们三个的效率都很高，信安的同学很快就完成了预计的开发任务，我也投入了大量的时间和精力，保证了自己的开发进度，黄同学则为我们提供了前后端设计时需要的数据库数据、前端页面的内容，并为我的设计提供了宝贵建议，协助我设计css（因为我的审美一开始确实有点糟糕），并且每次会议后都会记录会议纪要，为我们提供了较多的帮助。（其实本来分工是由她负责前端我负责后端的，但可能我时间比较多，开发后期也主要是实现前端功能，所以就抢了前端的工作，相应的她做了其他工作），我们四个同学之间遇到问题和有需求基本都是微信和qq私信联系，需求的同步和解决比较及时（比如需要什么样的猫咪数据格式，就会让黄同学帮忙整理，我这边需要部署服务器，就需要许同学那边放开端口等等），不过留学生今关叶月同学没有主动参与到我们的开发中，还是比较遗憾的。最后整合的时候大家沟通较多，解决了各种整合时的问题，将整个项目的前后端合璧，再和我的在线聊天平台合璧，展示前也做了比较充分的准备，比如一起准备提交材料、我和许同学设计ppt内容再由黄同学设计排版风格等。

总而言之，在本次团队合作中，我们小组一开始线下例会时的合作和学习氛围比较强烈，大家进度也比较统一，知识分享和交流都很有效，但当正式开发后，受到疫情影响从框架学习开始有同学逐渐掉队了，而我们三个开发的同学因为基础比较好（许秦）或者投入了比较多的精力学习（我），所以基本上就是自己一个人包了所有的开发任务，后续遇到的bug也都是自己解决（因为大家都在开发不同的功能，对彼此遇到的bug不熟悉，交流也很难获得有效的解决办法，不如直接查询网络），在最后整合的时候大家也共同努力，合作愉快。在初期和末期我们的团队合作做的很好，但是中间开发受疫情影响交流比较少，不过我们团队分工很明确，每个开发的同学也保证了自己的进度，因此完成了任务。

经过大家的努力，最后完成度还是比较高的，基本上没有bug，甚至完成了许多的优化设计，如图片懒加载、响应式布局等等，并且我们三个人在三台服务器上都分别实现了自己的部署，许秦同学还用restful api解决了跨域问题，我也用nodejs解决了websocket服务器部署时的跨域问题（另一个专注开发聊天室的小组却没有解决该问题hh），对于我们小组最后取得的技术成果，我还是比较满意的。此外，我非常喜欢我们小组的氛围，大家在线下例会时最喜欢“唠嗑”，我们会谈论学习、生活、实验室、实习、未来的焦虑，等等等等，大家也比较容易达成统一的意见，是非常有利于合作的氛围。可惜疫情的影响，外加专业和课余时间不同，让我们小组的合作并没有充分发挥潜力。

1. 特色工作

由于我主要负责开发，因此围绕本课程需要的技术分享几点我自己的特色工作。

1.**项目部署**：

在本次项目中，我掌握了丰富的部署技巧，我用到的技术栈是Vue+node.js。

我首先尝试了华为云自动的构建、部署功能，参考的视频是https://www.bilibili.com/video/BV1vb4y1C7ta?spm\_id\_from=333.337.search-card.all.click

使用说明：在图4所示页面，依次点击编译构建（本质上是打包，如果使用nodejs，等效于npm run build），再点击部署（本质上是部署刚才打包好的dist目录），即可部署成功。



图4 华为云自动化部署，可以云上构建（打包）也可以本地构建

部署成功后如果无法执行，可以打开网页console查看报错，如果无法访问请检查（1：云服务器的安全组设置是否开放了你需要的端口，是否有防火墙2：vue2可以检查index.js中的host设置，vue3没有找到该设置，我尝试的过程中使用localhost也可以部署），如果访问了却无法加载路径请检查publicpath属性是否配置正确，网上的教程进程一刀切教大家改成”./”实际上可以根据需要修改为任意路径，最简单的方法是打开console看他请求的资源在云服务器的什么路径，再找到自己打包好的index.html检查资源加载路径是否正确，如果不正确可以修改index.js中的publicpath属性（vue3中没有index.js,需要手动在根目录下增加vue.config.js）甚至手动修改打包好后的文件（一开始我看着打包好后的文件就是一堆乱码，但对代码足够熟悉后，甚至可以用搜索直接修改打包好文件的js和css）。通过以上的部署，就可以上线大部分的网页了，例如我尝试的图书管理系统。

但是，这样的部署只能运行打包好后的dist文件，对于需要在终端执行的指令是无效的，例如我的后端服务器需要在终端运行node server.js，那么怎么办呢？

我一直被这个问题困扰，因为网上没有找到任何相关的教程，可能大家部署的时候很少遇到这样的问题，即在云服务器上运行一个js而不是打包好的文件。

我查阅了几乎所有华为云的指导文件，但没有任何的帮助，后来看到大部分的部署教程都是使用阿里云，我便将整个代码迁徙到了阿里云上。

阿里云相比华为云，没有一键生成环境的指令（华为云自动构建前可以生成需要的环境，例如我需要的是node.js，或Django）,我首先查询了网上的博客（如何在云上配置环境），但是网上的博客都是非常复杂的教程，例如用xshell上传下载好的nodejs压缩包，然后安装在云服务器里。千万别用这样的方法！！

有了华为云的使用经验，我的第一反应就是作为类似的公司，阿里云肯定也有较方便的配置环境的方法，终于我找到了一个阿里云的官方手册，通过该手册，我成功配置了nodejs的环境。手册上有一个例程，即写一个js文件并通过公网ip访问来测试nodejs环境。我顿时灵光一线，这不正是我需要的功能么？

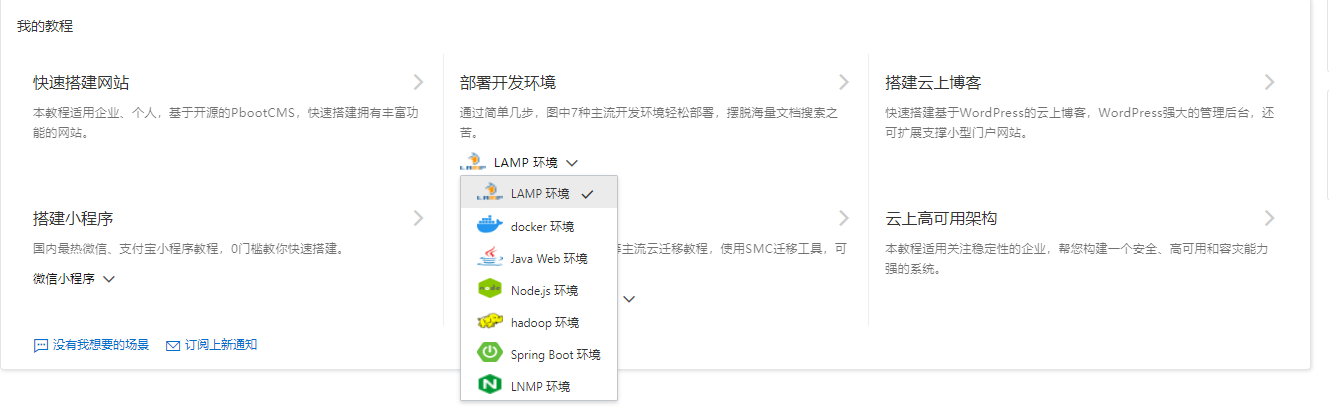


图5 阿里云官方文档，这几个都很实用（在弹性云资源的新版页面中才能看到）

但在实际上线时又遇到了一个新的bug，在本地运行的好好的node.js环境在云服务器上却无法执行了，从npm install开始便出现了很多报错。为此，我几乎检查了整个vue的框架，甚至一个个检查报错中提及的node\_modules文件下下的相关库@vue等，最终发现是因为package.lock.jason文件锁死了环境版本，而云服务器安装的node版本和webpack版本都与本地不同。于是我大胆删除了package.lock.jason并在package.jason文件中修改了安装的库的版本，最终成功安装，并且可以执行npm run serve和build指令了。

至于远程桌面的访问，很多教程教的是使用xshell和宝塔面板，经过我自己的摸索，我认为用xshell的方法已经过时了。我之前经常在做科研的时候访问实验室的电脑，所以我尝试用连接实验室二件套访问云服务器并且执行我打包好的文件，结果成功了！和xshell比方便许多，故此处做一些分享。

用到的软件：vscode+filezila（ide只要能远程ssh连接均可，如pycharm也可以）

使用方法：首先使用filezila，ssh连接云服务器，上传自己本地需要的文件，例如源代码、打包好的文件等等，filezila面板如下图所示：

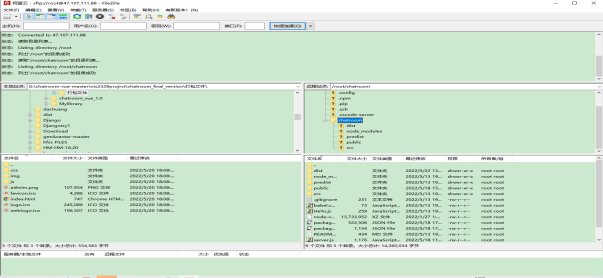


图6 filezila页面，和文件夹差不多

然后，在vscode点击左下角建立ssh连接，登录远程桌面，成功后如下图所示：

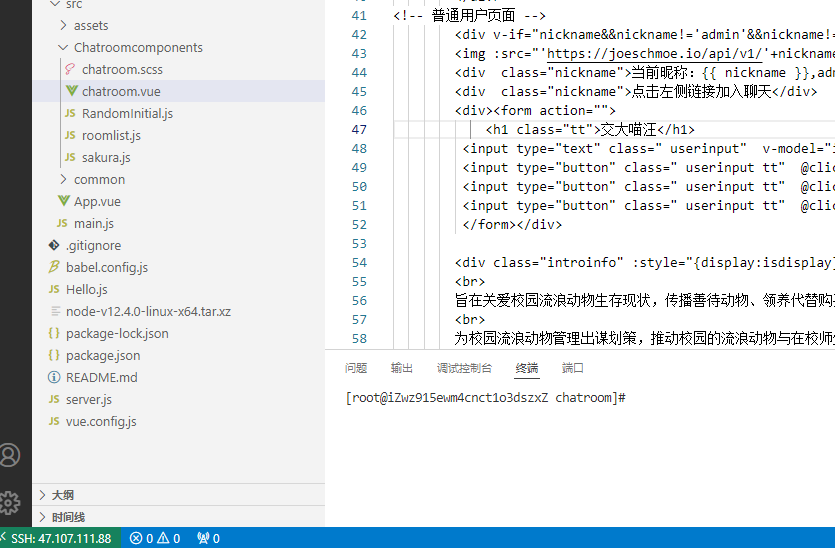


图7 远程连接后，可以用ide编程、终端运行（需要用tmux挂载）

然后就可以像在本地ide一样在云服务器自由修改代码和在云服务器的终端运行代码了！但是需要注意的是关闭ide云服务器的进程也会终止，因此要用tmux挂载进程，例如我分别挂载了服务器和客户端两个进程，在拆分bash或新建tmux端口后阿里云的环境变量会导致node环境消失，需要手动输入指令让当前终端进入node环境。

当然也可以切换到打包好的dist目录下执行live-server指令直接运行打包好的文件，这样加载速度会更快。该部署方法在华为云也同样可用，华为云环境配置比较方便。不管在哪个云服务器，**记得安全组开放端口**。

**2.团队合作方式的可能性改进**：

对于前端而言，测试可能不像后端那样需要写测试样例测试函数，而是希望自己的修改可以立刻展示在前端上。热更新机制即修改后立刻在前端显示，实现了这个需求，通过我上述的部署到服务器，便可以在服务器上同时多人修改css、vue、js文件，并且一直执行热更新，这样负责前端的同学可以一直看到自己的更新，负责后端的同学也可以实时看到前端同学提交的改动，并根据改动调整相应的功能，非常适合多人同时修改前端页面，或者前后端同时合作编程时的场景。我在项目后期也和黄同学尝试了这种方式，证明该方法是可行的，但如果能早点发现这样的操作方法，就不用后端的同学做完后等前端的同学设计，前端的同学更新需求后再反馈给后端了，可以提高很多的效率。

本质上而言，这是一种持续部署的方法。这种（vscode+filezila）的部署方法是我把实验室电脑的连接方法用在网页部署上的一种尝试，网上目前没有看到类似的教程（当然本质上和xshell还是很接近的），是我在充分了解了部署原理后按自己的理解尝试技术并且成功了，也算是我一个不小的收获。此外，我第一次挑战了看网上的教程和github源码纯自己设计css（虽然是将各种风格的css组件混拼起来用），弥补了前端课程中从来没有自己设计css的遗憾。考虑不同设备做了响应式布局，考虑到了服务器的传输带宽，做了图片渲染时的优化设计，运行打包好的文件将服务器的cpu占用从原先的30%降低到8%，有一些云服务器方面的收获。

**3.前端优化：**如懒加载、渲染时先渲染画质低的背景图再逐渐清晰、响应式布局等。

1. 总结与反思

我一直觉得做软件是一件快乐的事情，通过本次课程，我更加确信了这一点。现在回首这个学期的经历，我深刻的感受到了老师所说的“要学游泳，就钻进水里”的意思，这整个学期我学习了大量的前后端知识，完成了两个DjangoDemo的代码，这些都是我自己的尝试。我也系统的学习了专业知识，在前端课程中学习了很多的设计方法、优化思想，将这些知识又灵活运用到软件工程设计中，最终实现了我在大一下就开始一直想做的一个聊天室（虽然最后是在线交流平台），满足了我的开发欲。在整个开发过程中，我先是尝试了各种未知的技术，然后又将自己的一些专业知识、科研知识融入其中，花费大量的时间和精力钻研了许多技术，感觉自己的收获真的非常的大。真的非常感谢老师提供的这个宝贵的机会，让我有推自己一把的动力。这个学期中我经常是早上起来想着要实现什么功能，然后中午写完了初步的函数，接着在屏幕前debug和改前端样式就是半天，但最后战胜了bug的喜悦，总是让我觉得自己的努力很有价值。

虽然这门课程整体上是让我收获很多的，但我也有一些遗憾，反思首先是对我们小组合作的反思，我想造成我们小组分工不均的主要原因首先是因为大家一开始完全不熟悉，也可能是因为我们三个开发的同学都觉得自己就能完成大部分的工作，然后很少将未完成的任务交给另外两位同学做，导致实际上是三个人做主力的开发，这样的行为是不对的，应该更相信自己团队内成员的能力（可实际上我感觉确实自己开发会比较有效率，当我想把功能交给另一名队友实现时，我会非常担心ta会不会做了一些修改造成bug导致我要回退很多，于是最后就自己干了）。

我觉得还有一个问题是大家在选修这门课程时的动机是不一样的，对于信息安全专业的同学而言，这门课是必修课，自然需要全力以赴，对于非信息安全专业的同学而言，有像我这样一开始就抱着来学技术，并且有很多空余时间的大三同学，也有一些只是混交叉学分，或者抱着了解软件开发流程来的同学。这样不同动机的同学聚集在一起，自然就会导致开发时任务分工不均。但说实话作为团队内的一名普通队员，我即便意识到了这样的问题，却没有权力也没有办法去强迫其他同学一起努力，因此只能选择自己一个人多干一些。我觉得根源可能是在一开始立项的时候我们没有树立共同的团队目标，大家都是打算走一步看一步，因此遇到突发情况（疫情）节奏便不一致了。另外课业压力也是很大的影响因素，例如大四的日本留学生今关叶月同学好像就是因为要忙毕业论文所以参与开发很少，她本身是个很热情的同学，经常参与我们的讨论。作为一个团队，需要有人为大家在早期就树立共同的目标，并鼓励大家在懈怠时一直努力，是我学到的一个经验教训。作为交叉课程选修该课程的同学，真心希望你们把握住这个提升自己的机会。

最后最后，是我在展示时暴露出的问题，在整个开发过程中，我过于专注尝试不同的技术，学一些其实没什么用只是看起来很炫的代码（比如前端的毛玻璃设计），而忽视了最基本的需求，在项目展示的时候也有点主次不分明。作为一名希望从事算法设计和软件行业的同学，我不应该忘记所有的算法和软件都是服务于需求的（这一点字节的算法工程师也和我讲过），这是我学到的宝贵教训。

1. 致谢

谢谢马颖华老师的软件工程课程教学、进度督促和最后的点评教导，让我了解了许多软件工程开发的规范和流程，了解了软件工程师应该有的素养，并且在实践中让我挑战了自我，全方面锻炼了自己的能力，也学到了很多宝贵的教训。也谢谢我的组员们，我们一起留下了热情友好的合作回忆，希望大家在未来的专业学习中一帆风顺。

附：在线交流平台项目完整源码

<https://github.com/sjtuLLWWTT/NISSE2022>

Django框架学习，可以掌握基本Django框架和数据库，该教程有2.0版本（微信公众号中有源码和Django中文手册）

https://www.bilibili.com/video/BV1su411Z72Y?spm\_id\_from=333.337.search-card.all.click

跨域问题的非常详细说明（我用的是Node正向代理）：

https://blog.csdn.net/ch834301/article/details/119582822