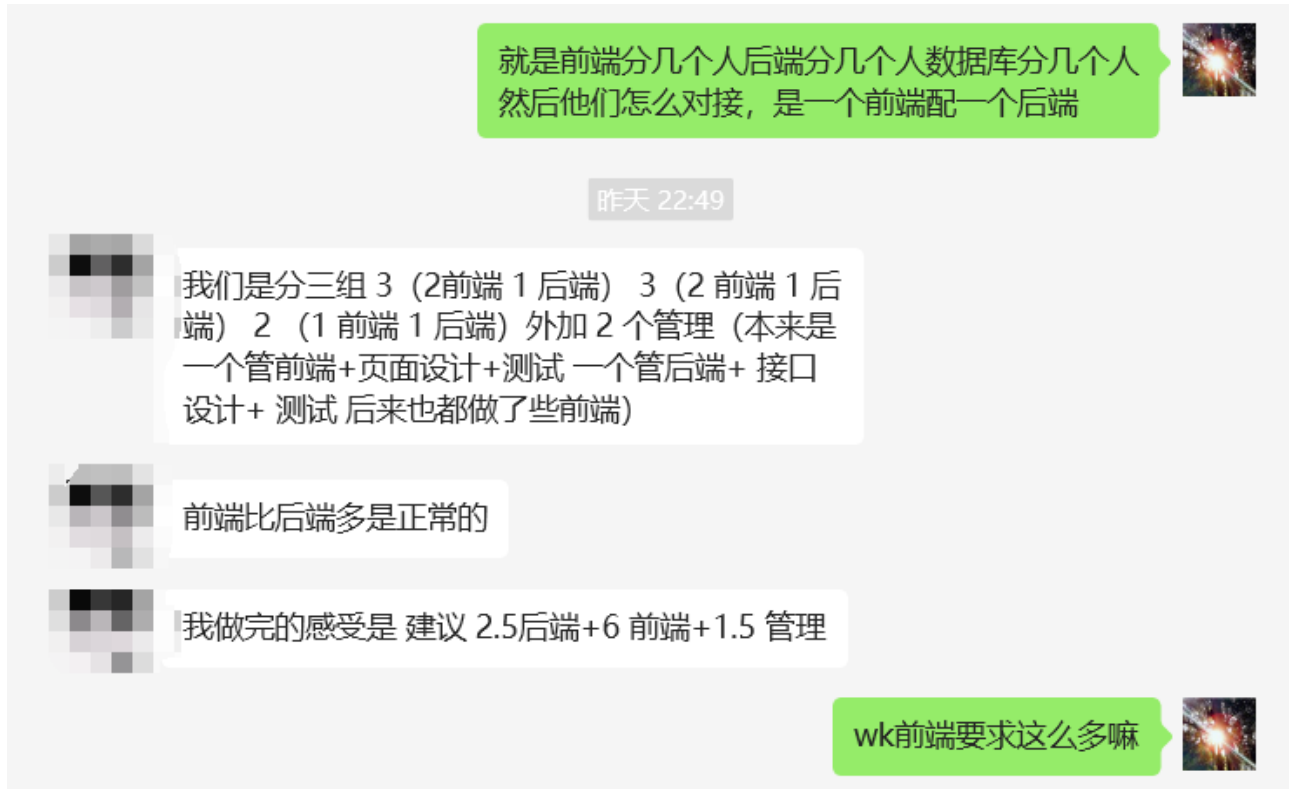


1. 讨论数据库如何设计（依照学长的项目） [农场项目](#)



有的图片过期了 将就看吧 () 我觉得还是一步一步来 比如说先把表设计好了, 页面设计好, 前端可以先安装一下所用的工具 跟 b 站学起来, 后端也是可以先本地跑跑, 随便测几个接口, 不着急写真正项目的内容

3. 人员怎么配置（怎么合作，避免前后端合不起来）




4. 前端学vue&node js，后端学C#，负责构建项目的同学把讨论结果以流程图的方式提供给前/后端同学，以便代码实现

5. 项目评分如下

数据库得分要点

- ☐ 表的数目；逻辑的复杂性；数据库的数据量大小（爬虫 or ChatGPT）《鼓励内卷》
- ☐ 主体功能的完善性：强调业务逻辑的完整与鲜明。比如一个图书馆的项目，没做研习室的预约被给了低分。
- ☐ 页面设计的美观性：非常非常强调这点，基本评语开头是界面美观的，非优即良
- ☐ 用户交互的自然性：很强调这一点，强调交互符合现实场景，比如用户手动输入IT

交互付出现实场景，比如用工厂代码输入订单号，会被认为是违反现实逻辑的交互；比如用扫码枪，被认为是很好的实现。

- ☐ 外部接口的调用与实现：偏爱酷炫的功能与实现，比如我们的系统里 地图接口，支付接口，物流接口等等如果能调用，会成为加分点。
- ☐ 高级可视化与统计：多种可视化图表、多维度分析，可能成为加分点
- ☐ PL/SQL 编程：问了助教 gg，他也不太理解；但每组的评语都会提及如“未使用 PL/SQL 编程”，估计使用了过程化 SQL 语言会加分。
- ☐ 代码的书写规范与严谨性：是会看代码的，从评语来看，注重数据库/函数的封装性，以及一些细节如“字符串的连接直接进行了”（个人不太理解，助教说是他写死了，应该用 connect 函数之类的）
- ☐ 代码仓库的安全性：***不要透露我们的仓库  给任何人*** 往届有答辩被恶意攻击的，还有得了优被其他同学顺着答辩链接扒了源码，找到部分抄袭依据，大幅降档的。

□ 评分依据：没有一个固定的做到什么程度就能优的标准，主要是看横向对比，请大家在答辩之前不要透露项目的进展情况、页面设计等等。

□ 玄学：老师答辩的心情、老师对我们的平时印象等等。

注：上述内容为和助教 gg 交流的一些总结，因涉及一些不可公开的往届信息，请勿外传



6. 加分与扣分

- 6.1 允许摆烂**2次**，即两次 相邻开会间不推进项目
- 6.2 允许开会时人不在**2次**，但是别人替你汇报
- 6.3 上述机会用完后，**不接受任何形式请假，直接扣成绩**
- 6.4 代码需要经常返工，扣分(**因需求改变导致的反工不算**)
- 6.5 代码复用性强，结构清晰 加分
- 6.6 在完成自己的工作后辅助别人完成工作 加分

4.7号前的目标

- 1. 所有人配置好Oracle数据库
- 2. 前端人员依据这个“FrontEndDevelop”学习并配置vscode [前端开发](#)
- 3. 后端开发人员根据“.net开发文档”学习并配置vs2022 [后端开发](#)
- 4. 确定项目布局与人员分配
- 5. 确定数据库设计思路

4.7日程