**1.项目技术分类**

项目开发中，所以用到的前端技术分为以下几类：

* 脚本语言，如：JavaScript
* 样式语言，如：CSS、Sass
* 框架，如：Vue3
* 组件库，如：Vuetify
* 构建工具，如：Vite3
* 其它工具库，如：axios

后端技术分为以下几类：

* 脚本语言，如：Python
* 框架，如：Django

**2. 脚本语言**

* **JavaScript**：浏览器的默认脚本语言
* **python**：浏览器的默认脚本语言

**2.1 JavaScript：**

不需要编译器，因为web浏览器使用HTML解释它；

JS可以跨多个浏览器、平台等工作；

使网站更具互动性，吸引访问者的注意力；

它比其他编程语言更快、更轻量级；

**2.2 python：**

Python适用于网站、桌面应用开发，自动化脚本，复杂计算系统，科学计算，物联网，游戏设计，机器学习，自然语言处理等很多方面。而且，既使对于那些从没有开发经验的人来讲，Python的代码也是简洁易懂的。由于Python程序代码简单，所以和与其他程序语言相比，后期的程序维护更容易，更舒心**。**

**3. 样式语言**

* **CSS**：浏览器默认的样式语言
* [**Sass**](https://links.jianshu.com/go?to=https%3A%2F%2Fsass.bootcss.com)：Sass 是一款强化 CSS 的辅助工具，它在 CSS 语法的基础上增加了变量 (variables)、嵌套 (nested rules)、混合 (mixins)、导入 (inline imports) 等高级功能，这些拓展令 CSS 更加强大与优雅。使用 Sass 以及 Sass 的样式库（如 Compass）有助于更好地组织管理样式文件，以及更高效地开发项目。

**选择理由：**

* Sass : 在安装项目依赖时，经常因为 Sass 相关的依赖而安装失败，Sass基于Ruby语言开发而成，因此安装sass前需要安装Ruby，而国内安装 Ruby 需要 更改镜像源，而这不是前端技术栈满园内的东西；而 Stylus 是 NodeJS 项目，不存在 Sass 这样的问题；

**4. 框架**

**4.1. Web框架**

* [**Vue**](https://links.jianshu.com/go?to=https%3A%2F%2Fcn.vuejs.org%2Fv2%2Fguide%2F) : 一套用于构建用户界面的渐进式框架。Vue是一款流行的JavaScript框架，可以帮助开发者构建用户界面。Vue具有简洁的语法和易用的API，提供了许多方便的功能，如数据双向绑定、组件化开发、虚拟DOM等。使用Vue可以快速构建响应式的前端应用程序,它也具有丰富的生态系统和强大的社区支持。提供了各种丰富的插件和组件库。使用Vue.js可以创建出具有良好用户体验的交互式前端应用。

**选择理由：**

* 轻量级的框架
* 双向数据绑定
* 组件化开发
* 单页面路由
* 学习成本低
* 虚拟dom
* 渐进式框架
* 数据和结构的分离
* 运行速度快
* 插件化

**4.2. Django框架**

Django具有以下特点：

* 功能完善、要素齐全：该有的、可以没有的都有，自带大量常用工具和框架，无须你自定义、组合、增删及修改。
* 完善的文档：经过十多年的发展和完善，Django有广泛的实践案例和完善的在线文档。开发者遇到问题时可以搜索在线文档寻求解决方案。
* 强大的数据库访问组件：Django的Model层自带数据库ORM组件，使得开发者无须学习其他数据库访问技术（SQL、pymysql、SQLALchemy等）。
* 灵活的URL映射：Django使用正则表达式管理URL映射，灵活性高。新版的2.0，进一步提高了URL编写的优雅性。
* 丰富的Template模板语言：类似jinjia模板语言，不但原生功能丰富，还可以自定义模板标签，并且与其ORM的用法非常相似。
* 自带后台管理系统admin：只需要通过简单的几行配置和代码就可以实现一个完整的后台数据管理控制平台。
* 完整的错误信息提示：在开发调试过程中如果出现运行错误或者异常，Django可以提供非常完整的错误信息帮助定位问题。

**5. 组件库**

Vuetify是一个强大的vue组件框架，从头开始构建，易于学习和掌握。我们的UI组件集合通过足够的自定义选项在整个应用中保持一致的样式，以满足任何用例。

优点：

功能丰富：Vuetify完全按照Materail设计规范进行开发，每一个组件都经过精心设计，具有模块化、响应式和优秀的性能。使用独特和动态的布局，可以自由的定义应用程序并使用SASS变量自定义组件的样式。

**6. 虚拟机**

虚拟机（Virtual Machine）指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。在实体计算机中能够完成的工作在虚拟机中都能够实现。在计算机中创建虚拟机时，需要将实体机的部分硬盘和内存容量作为虚拟机的硬盘和内存容量。每个虚拟机都有独立的CMOS、硬盘和操作系统，可以像使用实体机一样对虚拟机进行操作。

**文件管理为什么用Django比较方便**:

Django是一款功能强大的Web框架，专注于快速开发和高效的开发体验。文件管理系统通常涉及到用户管理、文件上传、权限控制等复杂的后端功能。Django提供了许多内置的功能和工具，如用户认证系统、模型和数据库管理、表单验证、权限管理等，使开发文件管理系统更加方便。Django还具有良好的安全性和稳定性，提供了许多可扩展的插件和第三方库，可以满足复杂的文件管理需求。

**模型的继承Django和flask的区别:**

在Django中，模型的继承功能非常强大。Django使用了基于表的继承模型，其中一个模型可以从另一个模型继承属性和方法。这种继承方式允许在模型层面上实现抽象和多态的概念，使得模型的设计更加灵活和可复用。在Django中，你可以通过使用抽象基类、多重继承、Proxy模型等来实现模型的继承。

而在Flask中，模型的继承相对简单。Flask是一个轻量级的框架，没有内置的模型层或ORM工具。通常情况下，开发人员可以根据自己的需求选择适合的ORM工具，如SQLAlchemy，来处理模型的继承。SQLAlchemy提供了更灵活的ORM功能，并且可以与Flask无缝集成。

**校验为什么用Django而不用flask:**

Django和Flask都提供了校验数据的机制，但Django在这方面具有一些优势。

Django内置了强大的表单验证和模型验证功能。可以使用Django的表单来定义字段、校验规则以及错误提示信息，然后进行数据的校验和清理。Django的模型验证机制也可以通过定义模型的字段选项和自定义验证器来校验数据的有效性。

与此相比，Flask没有内置的表单或模型验证功能，开发人员需要选择适合的验证库，并自行实现数据的校验逻辑。虽然可以使用第三方库，如WTForms来实现表单验证，但它与Flask的结合可能需要一些额外的工作。

Django提供了更强大和方便的数据校验功能，适用于构建复杂的表单和模型，并且更容易上手。而Flask则更加灵活和轻量，需要更多的自定义工作来实现数据的校验。