网络应用开发课程设计

题目

**组员信息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 姓名 | 学号 | 成员贡献 | 课程设计成绩 |
| 1 | 曾子骏 | 2021212237 | 独立完成个人主页搭建 |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

Github代码仓库URL：https://github.com/00zzj00/00zzj00.github.io

# **引言**

****决定独立完成课程设计。我选择搭建自己的个人主页，一方面考虑到这项任务自主性比较强，有助于独立完成任务并学习相关知识，另一方面考虑到自己完成作业后还可能将作业内容用到之后的学习生活工作中。****

# 功能与设计

## 功能介绍

使用git、hexo搭建个人主页（blog）

## 系统设计

系统运行的逻辑和关键设计：

**版本控制：** Git用于版本控制，允许你跟踪项目的变化、协作以及回溯到不同的版本。

**静态网站生成：** Hexo是一个静态网站生成器，它将Markdown等格式的源文件转换成HTML、CSS和JavaScript等静态文件。用户编辑Markdown文件，然后使用Hexo生成静态网站。

系统架构：

**前端：** 由Hexo生成的静态文件，包括HTML、CSS、JavaScript等。这些文件可以托管在任何静态文件托管服务上，如GitHub Pages、Netlify等。

**后端：** 由于是静态网站，通常没有后端服务。Git服务器（如GitHub）用于存储源代码和发布生成的静态文件。

逻辑流程：

用户编辑个人主页内容，通常是通过Markdown文件。

使用Git进行版本控制，将更改提交到版本库。

Hexo通过命令行工具执行生成静态网站的操作。

生成的静态文件可以被托管在任何支持静态文件的服务上，例如GitHub Pages。

通信协议：

**Git通信协议：** Git使用HTTP、SSH等协议进行通信，允许用户推送（push）和拉取（pull） 代码。

**Hexo本身不涉及通信：** Hexo用于本地生成静态文件，通信主要涉及到使用Git将代码推送到托管服务或拉取最新的代码。

# **功能实现**

关键技术：

**（1）Git：**

版本控制：Git是一个分布式版本控制系统，用于跟踪文件的变化，协作以及管理项目的不同版本。

分支管理：允许并行开发和测试不同功能的分支，并将它们合并到主分支中。

**（2）Hexo：**

静态网站生成器：Hexo基于Node.js，可以将Markdown等格式的文档转换为静态网页，生成包括HTML、CSS和JavaScript等文件的静态网站。

主题和插件：允许用户根据需求选择不同的主题和插件，定制个人主页的外观和功能。

**（3）Markdown：**

用于编写内容的简单易读易写的文本格式，Hexo支持Markdown作为内容源文件的格式。

**（4）HTML/CSS/JavaScript：**

前端技术：生成的静态网站由HTML、CSS和JavaScript构成，定义了网页的结构、样式和交互。

**（5）Web服务器和托管服务：**

GitHub Pages、Netlify等。这些服务提供静态文件的托管和部署，让用户能够将生成的网站部署到互联网上。

**（6）Node.js：**

Hexo是基于Node.js构建的，因此对于扩展或定制Hexo的功能，理解Node.js也是有益的。

# **功能展示与评测**

系统可用性和完整性

# **总结**

总结与问题