**MyHomePage项目**

**概要设计说明书**

目录

[1. 引言 3](#_Toc183174835)

[2. 系统架构设计 3](#_Toc183174836)

[2.1. 总体架构 3](#_Toc183174837)

[2.2. 技术栈 3](#_Toc183174838)

[3. 系统功能设计 4](#_Toc183174839)

[3.1. 网址导航功能 4](#_Toc183174840)

[3.1.1. 功能描述 4](#_Toc183174841)

[3.1.2. 主要功能模块 4](#_Toc183174842)

[3.1.3. 数据存储 4](#_Toc183174843)

[3.2. 日程管理功能 5](#_Toc183174844)

[3.2.1. 功能描述 5](#_Toc183174845)

[3.2.2. 主要功能模块 5](#_Toc183174846)

[3.2.3. 数据存储 5](#_Toc183174847)

[4. 数据流与交互设计 6](#_Toc183174848)

[4.1. 前端交互设计 6](#_Toc183174849)

[4.2. 后端交互设计 6](#_Toc183174850)

[5. 部署方案 7](#_Toc183174851)

[5.1. 服务器部署 7](#_Toc183174852)

[5.2. 数据存储与管理 7](#_Toc183174853)

[6. 安全性与隐私 7](#_Toc183174854)

[7. 总结 7](#_Toc183174855)

# 引言

本项目旨在实现一个简单的网页应用，提供网址导航和日程管理功能。该应用前端通过HTML、CSS和JavaScript实现，使用Bootstrap和jQuery框架，后端使用Node.js处理部分业务逻辑。应用部署在局域网内的OrangePi服务器上，使用Nginx进行反向代理和服务提供。信息存储采用Cookie技术，避免了数据库的使用。

# 系统架构设计

## 总体架构

系统采用前后端分离的架构，前端负责界面展示与用户交互，后端处理用户请求与数据管理。服务器部署在局域网内的OrangePi上，使用Nginx提供静态文件服务，Node.js用于处理部分动态请求。

* 前端：HTML、CSS、JavaScript，结合Bootstrap框架进行响应式设计，使用jQuery处理DOM操作和AJAX请求。
* 后端：Node.js实现轻量级服务器功能，处理用户的请求（如添加、删除书签和日程），并通过Cookie存储用户数据。
* 数据存储：使用Cookie存储书签和日程信息，避免了数据库的使用。

## 技术栈

前端：

* HTML5、CSS3、JavaScript
* Bootstrap 4
* jQuery

后端：

* Node.js
* Express框架

服务器：

* OrangePi 作为服务器
* Nginx 作为反向代理服务器

# 系统功能设计

## 网址导航功能

## 功能描述

用户可以在页面上快速管理常用网址，通过书签导航访问常用网站，支持添加、删除书签。

## 主要功能模块

书签管理：

* 添加书签：用户可以手动输入网址并添加书签，信息保存在Cookie中。
* 删除书签：用户可以删除已保存的书签，Cookie中的相关数据会被清除。

书签访问：

* 访问书签：点击书签图标后，会跳转到对应的网址。

## 数据存储

Cookie：书签数据以JSON格式存储在Cookie中，包括每个书签的URL和域名信息。用户关闭浏览器后数据仍能保留。

## 日程管理功能

## 功能描述

用户可以通过日程管理功能创建、修改和删除个人日程，同时支持日、周、月视图的转换，方便管理个人事务。

## 主要功能模块

日程管理：

* 创建日程：用户可以添加新的日程，设定日期、时间和描述。
* 移动/修改日程：用户可以修改已创建的日程内容。
* 删除日程：用户可以删除不再需要的日程。

视图转换：

* 转换月/周/日视图：用户可以选择不同的视图模式（例如月视图、周视图、日视图），以不同方式查看日程。

## 数据存储

Cookie：日程信息存储在Cookie中，每个日程包括日期、时间、描述等字段，支持简单的增、删、改操作。

# 数据流与交互设计

## 前端交互设计

前端页面通过HTML和CSS构建，用户通过点击按钮和输入框与系统交互。主要交互逻辑包括：

* 用户通过输入网址和点击添加按钮来添加书签。
* 用户点击书签图标访问相应网站。
* 用户通过输入日程信息并点击按钮来添加日程。
* 用户通过切换按钮选择不同的日程视图。

## 后端交互设计

暂无。

# 部署方案

## 服务器部署

* 硬件环境：使用OrangePi作为服务器，配合Nginx和Node.js运行。
* 软件环境：在OrangePi上安装Node.js和Nginx，Nginx负责反向代理，Node.js处理API请求。
* 部署方式：通过局域网进行访问，服务器没有公网IP和域名，用户可以通过局域网中的IP地址访问该网页。

## 数据存储与管理

Cookie存储：前端通过JavaScript操作Cookie存储书签和日程数据。每个用户的书签和日程数据在其浏览器中保存，不需要额外的数据库。

# 安全性与隐私

* 数据存储：由于使用的是Cookie存储数据，数据保存在用户的浏览器中，不会泄露给其他用户。
* 隐私保护：所有存储的数据仅限于当前用户使用，且不会上传至服务器，确保用户隐私不被侵犯。

# 总结

本系统通过简单的前后端架构提供了网址导航和日程管理功能。使用Node.js进行后端处理，前端通过Cookie存储数据，采用Bootstrap和jQuery实现响应式界面，确保用户能够方便地管理个人网址书签和日程。项目简洁、轻量，并且部署在局域网内的OrangePi服务器上，可在局域网下使用。