# 项目名称：远程服务器管理系统

## 1. 项目概述

本项目是一个基于 Flask 和 SQL Server 的服务器管理系统，主要功能包括：用户注册、登录、管理各种数据表格（用户信息表、操作记录表、文件编辑记录表、服务器信息表、端口转发规则表）的增删改查操作，以及远程操作服务器的批量连接、控制台命令执行、文件传输、软件安装删除、调度等功能。

## 2. 技术架构

本项目使用 Python Flask 框架作为后端，使用 SQL Server 数据库管理数据，前端使用 HTML、CSS 和 JavaScript 实现。其中，Flask 提供了路由管理、请求处理、数据处理等后端基础功能，SQL Server 实现了数据存储与管理，而前端则提供了用户界面和交互。

## 3. 项目特点

* 可以在一个平台上集成多种远程服务器操作功能，提高操作效率和可靠性；
* 前端页面使用了Bootstrap框架，具有良好的视觉效果和交互性；
* 利用Flask-Caching进行缓存，可以提高页面加载速度；
* 利用Flask-Pagination进行分页，可以方便地管理表格数据；
* 支持多用户注册、登录、管理，保证系统的安全性；
* 所有的操作记录都会被记录到操作记录表中，方便管理人员进行查阅。

## 4. 功能实现

本项目实现了以下主要功能：

* 用户注册、登录：用户可以通过邮件进行注册，注册后可以登录系统，管理数据和操作服务器。
* 数据表格管理：包括用户信息表、操作记录表、文件编辑记录表、服务器信息表、端口转发规则表等数据表格，用户可以对这些表格进行增删改查操作。
* 远程操作服务器：支持批量远程连接服务器，执行控制台命令（如显示查询目录、查询主机名、查看进程、查询 CPU 占用率、查询内存占用率、查询网络IO、查询磁盘IO、杀进程、上传下载文件等）、文件传输、软件安装删除、调度等功能。

## 5. 技术细节

### 5.1 数据库设计

本项目中的数据表格使用 SQL Server 进行管理。具体的数据表格结构如下：

* 用户信息表：包含用户注册信息和登录信息。
* 操作记录表：记录用户在系统中的各种操作。
* 文件编辑记录表：记录用户对系统中的文件进行编辑的操作。
* 服务器信息表：记录服务器的基本信息。
* 端口转发规则表：记录服务器上的端口转发规则。

### 5.2 前后端交互

前端页面使用 HTML、CSS 和 JavaScript 实现，后端使用 Flask 框架提供 API 服务。前后端之间通过 AJAX 技术进行数据交互。

### 5.3 远程连接服务器

本项目使用 Paramiko 库实现远程连接服务器功能。通过 Paramiko 库实现 SSH 会话，并可以在会话中执行控制台命令，实现批量远程操作。

### 5.4 调度功能

在调度管理页面，用户可以添加调度任务。用户需要指定任务的名称、任务类型（间隔或定时执行）、间隔时间或执行时间、执行的命令和参数，以及任务所在的服务器。当任务被添加后，系统会自动启动该任务。

用户可以删除已有的调度任务。在调度管理页面，用户可以选择要删除的任务，点击“删除”按钮即可。

系统会自动记录调度任务的执行日志。用户可以在调度日志页面查看所有的调度任务日志。用户可以选择要查看的任务和日期范围，系统会自动查询指定任务在指定日期范围内的执行日志。

### 5.5 文件编辑记录

本项目记录了每个文件的编辑记录。当用户打开文件进行编辑时，系统会自动记录编辑时间、编辑者、编辑内容等信息。用户可以在文件编辑记录页面查看所有的文件编辑记录。

在文件编辑记录页面，用户可以选择要查看的文件，系统会自动查询指定文件的编辑记录，并将查询结果显示在页面上。

### 5.6 端口转发规则配置

本项目支持端口转发功能。用户可以在端口转发页面添加、修改、删除转发规则。用户需要指定源地址、源端口、目标地址、目标端口等信息。在添加或修改规则时，系统会自动检测目标地址和端口是否可用。

在端口转发页面，用户可以添加新的转发规则。用户需要指定源地址、源端口、目标地址、目标端口等信息。

#### 5.7 文件编辑功能

本项目支持远程编辑指定服务器上的文件。用户可以在系统中选择需要编辑的文件和服务器，并且可以使用文本编辑器进行编辑。编辑后，用户可以选择保存或者取消编辑。当保存修改后，文件将被上传到指定的服务器。

#### 5.8 软件安装删除功能

本项目支持在指定的服务器上安装或者删除软件。用户可以在系统中选择需要操作的服务器，并选择需要安装的软件包。安装完成后，系统会将操作结果返回给用户。用户通过输入要删除的软件名即可删除该软件。

## 6. 未来计划

1. 添加批量远程操作的进度条，提高用户体验；
2. 添加实时监控服务器状态的功能，如服务器负载、磁盘使用率、网络流量等；
3. 支持不同的数据库类型，如MySQL、Oracle等；
4. 支持邮件通知功能，如操作完成后发送邮件通知用户。

## 7. 总结

本项目旨在通过一个统一的平台，集成多种远程服务器操作功能，提高操作效率和可靠性。通过Flask框架、SQL Server数据库等技术实现了用户各种操作记录、文件编辑记录、服务器信息和端口转发规则的管理。主要功能包括邮件注册、登录、用户管理、表格管理、服务器管理、远程操作等。在远程操作方面，支持多种控制台命令，如显示查询目录、查询主机名、查看进程、查询CPU占用率、查询内存占用率、查询网络IO、查询磁盘IO、杀进程、上传下载文件、查询调度、自定义调度、删除调度、读取编辑文件、端口转发规则配置、软件安装删除等功能。所有这些功能都可以通过批量远程操作来实现。

该项目的部署非常简单，只需要在服务器上安装Python和SQL Server，并按照要求安装相关依赖库即可。开发人员可以使用PyCharm等开发工具进行开发和调试，而普通用户可以通过浏览器来访问Web页面，完成各种操作。

总体来说，该项目的优势在于集成了多种服务器操作功能，实现了一站式操作，同时也具备了良好的可扩展性和可维护性。