2024大创 - 技术文档

链接：

<https://github.com/Caspainnn/2024DAISO>在此网站上传代码

<https://sqlitestudio.pl/> sqlite可视化工具

<https://gitee.com/huinkl/2023-big-innovation-project>第二版核心代码整合

<https://github.com/Tora-chen/flask>数据库操作的代码

<https://shipinshu.site/> 两个简单的数据库操作

<https://www.bilibili.com/video/BV1L7411M7Ut/> Vue.js纪录片 - 中英CC字幕 - Honeypot呈现

[SQLite - Python\_SQlite中文网](https://www.sqlite.net.cn/tutorial/51.html)sqlite在python上实现，用这个就够了

[Flask-Migrate模块使用\_flask migrate-CSDN博客](https://blog.csdn.net/feit2417/article/details/86592319?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522172044248616800182772670%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=172044248616800182772670&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-6-86592319-null-null.142^v100^pc_search_result_base1&utm_term=%E6%97%A0%E6%B3%95%E4%BB%8E%E6%BA%90%E8%A7%A3%E6%9E%90%E5%AF%BC%E5%85%A5%E2%80%9Cflask_migrate%E2%80%9D&spm=1018.2226.3001.4187) 还写了一些关于文件结构的配置信息

目前开发进度：

语音转文字 √

视频转音频 √

视频转图片，图片筛选 √

-->优化：不用保存，直接比较删除 √（实现是实现了，但是感觉保留的太少了）

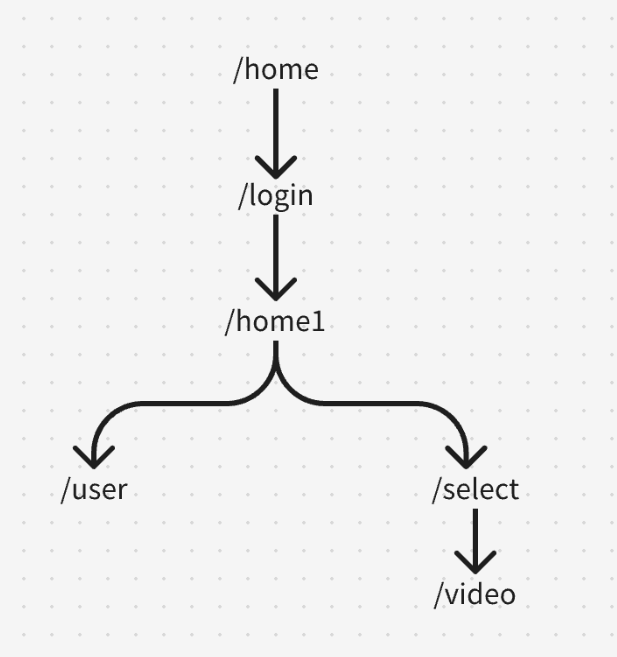
视频截图描述（不采用简单OCR，采用ChatGPT描述） √

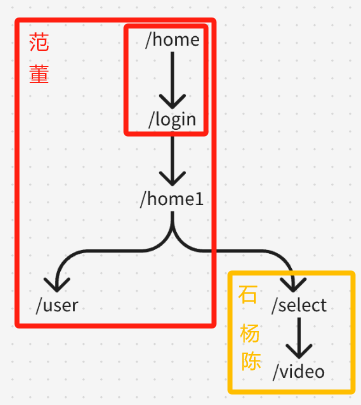
文件上传 √ （尚未部署）

数据库表设计

数据库操作设计

网页结构：





这个是对接分组，我想的是先把难啃的两个搞定后再去弄前面比较简单的。董倩这边也不要着急，我和志虎弄完后就和你一起去弄剩下的





**软件的主要功能：**

1.原视频转录成多栏截图展示

2.可进行翻滚查看视频课程

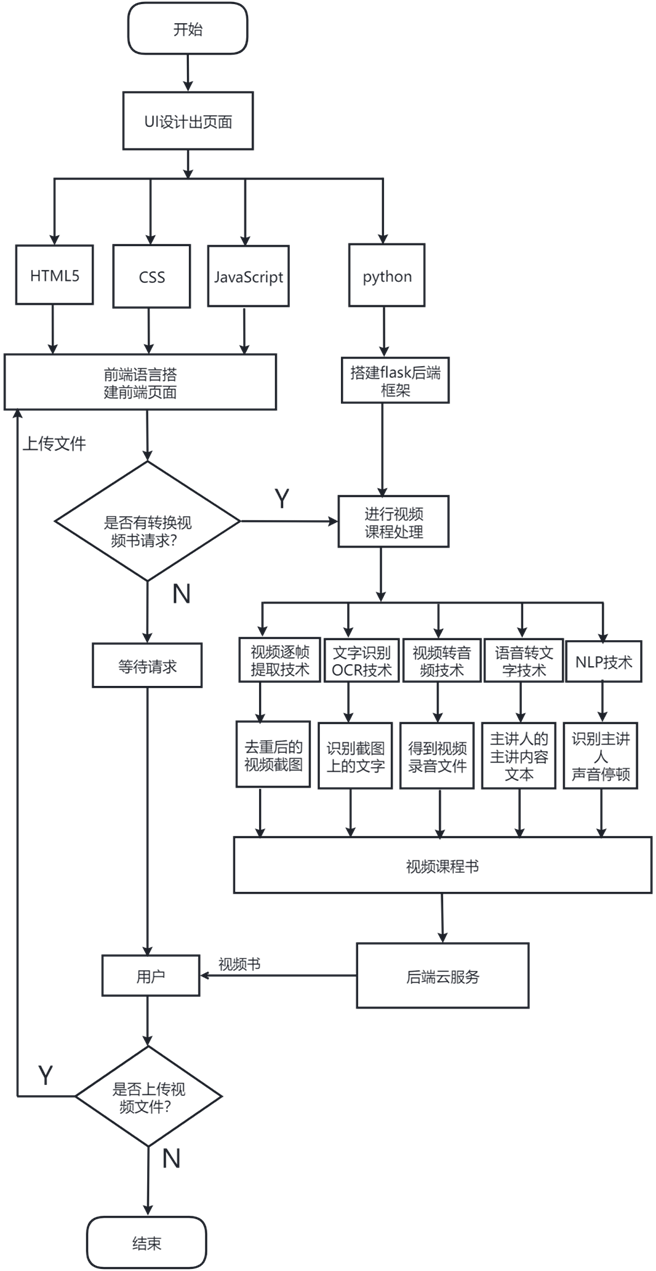
3.通过点击图片或文字可进行评论和标记

4.视频语音转换成文本

5.重塑视频信息流，便于搜索

1. 视频书自动生成目录，便于定位

**软件开发流程图**



技术栈：

VUE，Flask。数据库：SQLite.

部署在Azure上。

前端

目前任务：

页面布局，将网页的大概样子做出来（模仿coursera网站），有两个页面，一个页面是首页，另一个页面是视频播放的页面。

首页：介绍系统功能，设置按钮跳转视频播放页

视频播放页：字幕、笔记（注释分为对图片和字幕）、讨论、视频图片做成下拉菜单（分为单栏、双栏、三栏）

播放页笔记布局建议（from陈志虎）：



左框放置视频的截图，右框放置截图的OCR结果。

接口设计

* 前端发出课程id，后端返回该课程下所有视频的名称和id

请求：

{

"lecture\_id": 114514,

}

响应：

{

"vedios": [{

"vedio\_id": 1919810,

"vedio\_name": "1. 极限的概念"}，

{

"vedio\_id": 1919811,

"vedio\_name": "2. 极限的运算"},

]

}

* 前端发出视频id，后端返回视频的字幕文本、视频截图的id、视频截图的描述

请求：

{

"vedio\_id": 1919810,

}

响应：

{

"vedio\_info": {

"transcript": "大家好，我是宋浩老师.........本节课到此结束，谢谢大家！",

"images": [{

"image\_id": 12345,

"image\_desctription": "这张图描述了极限的基本概念"

}

{

"Image\_id": 12346,

"image\_desctription": "这张图描述了极限的运算方法”

}

]

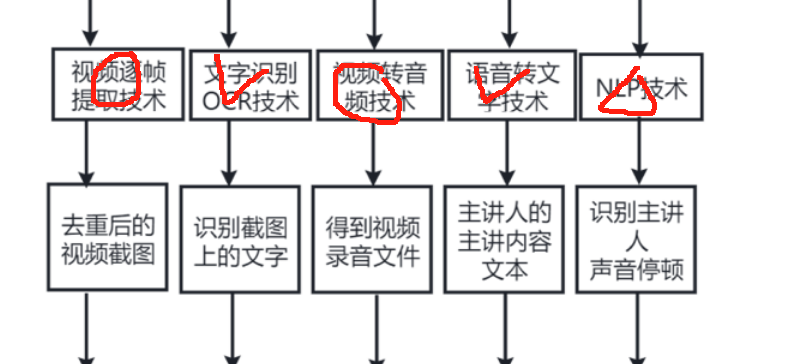
}

}

后端

目前任务：  
实现提及的基本功能，不用管数据库相关，先把功能实现再说。得到相应的结果后再考虑数据导入数据库。然后再实现渲染。

后端任务分配：Oyjl √czh 三角形dq



关于功能1：（已完成）

原视频转录成多栏截图展示

暂时放置，感觉对本项目实际作用意义不大

有个点，对于截屏的部分。相同的截屏需要删除，这是必须的。但是对于有些ppt讲解的内容，老师会在同一张ppt上勾勾画画，这个时候应该是属于相似的内容，应该也需要删除，否则在此基础上进行文字识别意义不大。

具体来说，本功能需要实现下面几个小功能：  
1. 间距截图（已完成）

1. 删除相同截图（已完成）
2. 删除相似截图（已完成）

反思，其实没必要实现2，因为相似一定大于相同，所以只需要删除相似的图片即可，这样肯定会删除相同的图片

深入思考：关于保留相似的图片，应该以后者为主，因为后者会包括前面的信息，差值是由于后者有更多的信息

功能2：（完成）

读入截图的id，将截图片的url发送给ChatGPT，ChatGPT生成图片描述。

关于功能3：（已完成）

视频转音频

就是写一个python文件，输入视频，输出音频

[Python实现视频转音频、音频转文本的最佳方法\_音频转文字 python-CSDN博客](https://blog.csdn.net/joe_g12345/article/details/136175031?ops_request_misc=&request_id=&biz_id=102&utm_term=%E8%A7%86%E9%A2%91%E8%BD%AC%E9%9F%B3%E9%A2%91python&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-4-136175031.142^v100^pc_search_result_base1&spm=1018.2226.3001.4187)

功能4：（完成）

读入一个音频文件，发送给OpenAI Whisper，将接收到的文本结果写入一个文本文件。

数据库设计

表：lecture,用于存储一个课程的id、名称、描述、封面图id

键：

|  |  |  |
| --- | --- | --- |
| lecture\_id | lecture\_name | description |
| 长整数类型 | 文本类型 | 文本类型 |

表：video，用于存储视频id、视频的名称、视频的字幕

键：

|  |  |  |  |
| --- | --- | --- | --- |
| video\_id | video\_name | transcript | lecture\_id |
| 长整数类型 | 文本类型 | 文本类型 | 长整数类型 |

表：note，用于存储视频截图id和该截图的描述(一对一关系)，和截图所属的视频的id。

键：

|  |  |  |
| --- | --- | --- |
| img\_id | description | video\_id |
| 长整数类型 | 文本类型 | 长整数类型 |

数据库操作

知识点：

**1.相对路径：**[相对路径](https://so.csdn.net/so/search?q=%E7%9B%B8%E5%AF%B9%E8%B7%AF%E5%BE%84&spm=1001.2101.3001.7020)：通常使用“.”或“..”开头。

“./”表示当前文件所在的目录，可以省略不写；

“../”表示当前目录的上一级目录；

“../../”表示上上级目录；

“/”开头代表[根目录](https://so.csdn.net/so/search?q=%E6%A0%B9%E7%9B%AE%E5%BD%95&spm=1001.2101.3001.7020)。

**数据库**

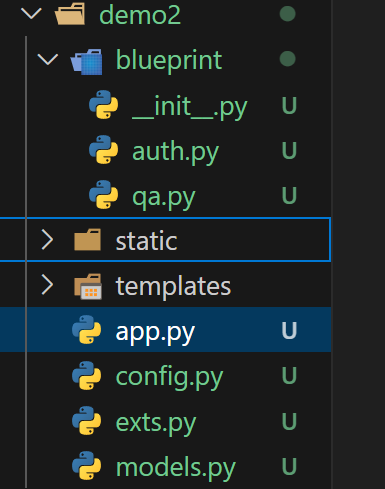
**SQLite**

**Flask**

**一个简单的html模板：**

下面是一个简单的HTML模板示例，它包含了一些基本的HTML结构元素，如<!DOCTYPE html>声明、<html>标签、<head>和<body>部分，以及一些常用的标签如<title>、<meta>（用于设置字符集和视口）、<h1>（标题）、<p>（段落）、<a>（链接）、<ul>和<li>（无序列表）等。

**Flask结构搭建**

****

**app.py**