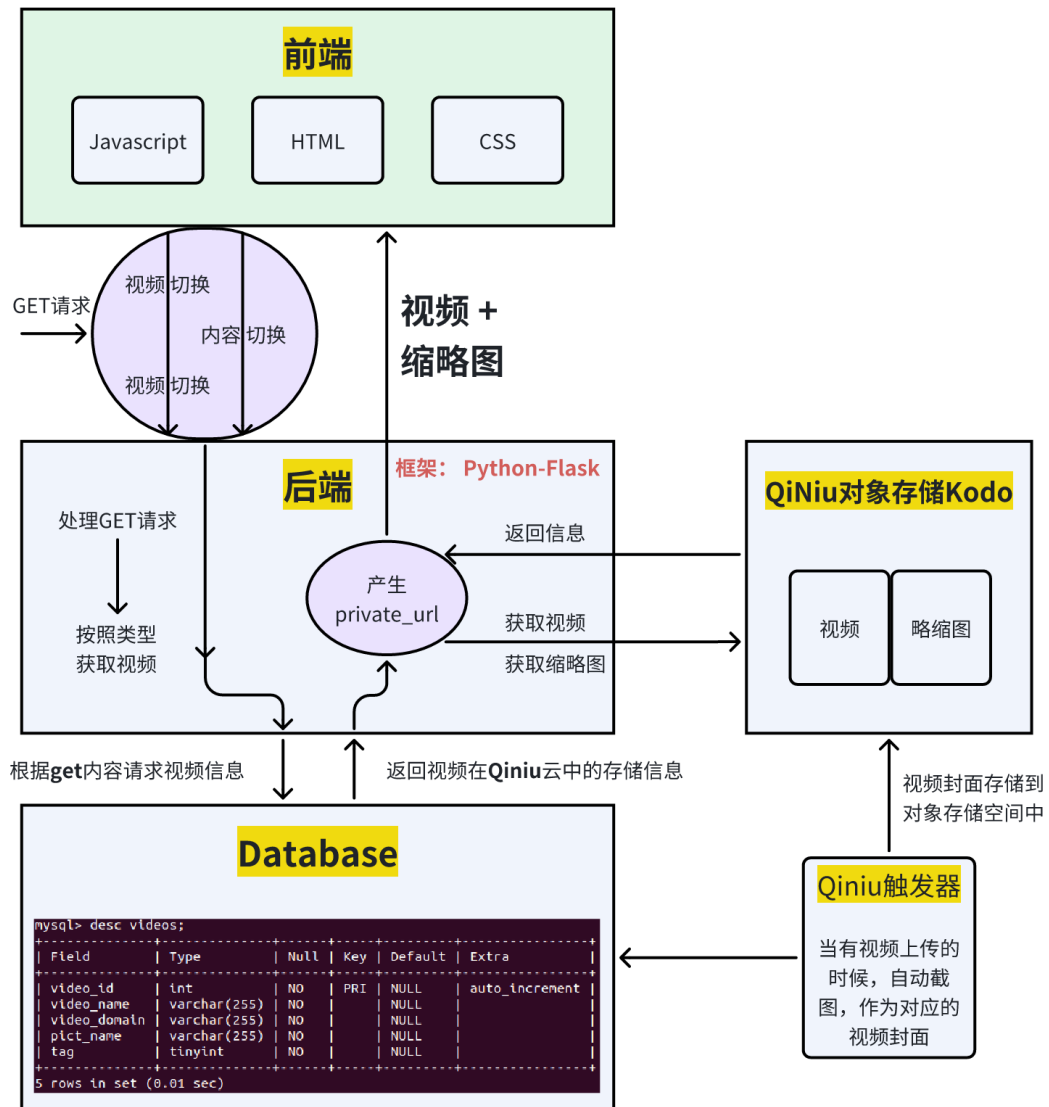


# 架构设计

## 一、架构设计图



## 二、各模块简述

### (1) 前端：

前端通过一些代码实现了 **抖音 Web 版** 的简单复刻，覆盖了题目要求的所有基础功能，并为拓展功能预留了一些接口。主要包含三个文件：

**index.html** 为网页的主文件，定义了页面的结构和内容；

**styles.css** 为网页的样式文件，定义页面的样式；

**script.js** 为网页的脚本文件，实现了页面的交互和动态功能。

### (2) 后端：

通过 Python 的 Flask 架构，覆盖了题目要求的基础功能。代码中主要包括 `/next_video`, `/pre_video`, `/fetch_date`, 等几种命令，接收到命令之后，执行对应的数据库请求，然后把获得的数据库连接保存到内存中，之后返回给前端。具体包括返回视频缩略图，返回视频，上一个视频，下一个视频，分类页面等功能。

### (3) Database：

在从网络上获取到视频素材之后，先进行分类，在上传至七牛存储之前，先根据视频分类将 **视频名称 略缩图名称 视频存储 domain 视频 id** 等内容存储到 Mysql 中，然后使用七牛云的上传 SDK 将视频上传至云存储中。对应不同的请求，后端通过不同的 SQL 语句，来获取不同分类的视频，从数据库中读取到对应的链接，然后返回对应的链接给前端即可。

下表是对应的数据库表配置信息：

```
mysql> desc videos;
```

Field	Type	Null	Key	Default	Extra
video_id	int	NO	PRI	NULL	auto_increment
video_name	varchar(255)	NO		NULL	
video_domain	varchar(255)	NO		NULL	
pict_name	varchar(255)	NO		NULL	
tag	tinyint	NO		NULL	

```
5 rows in set (0.00 sec)
```

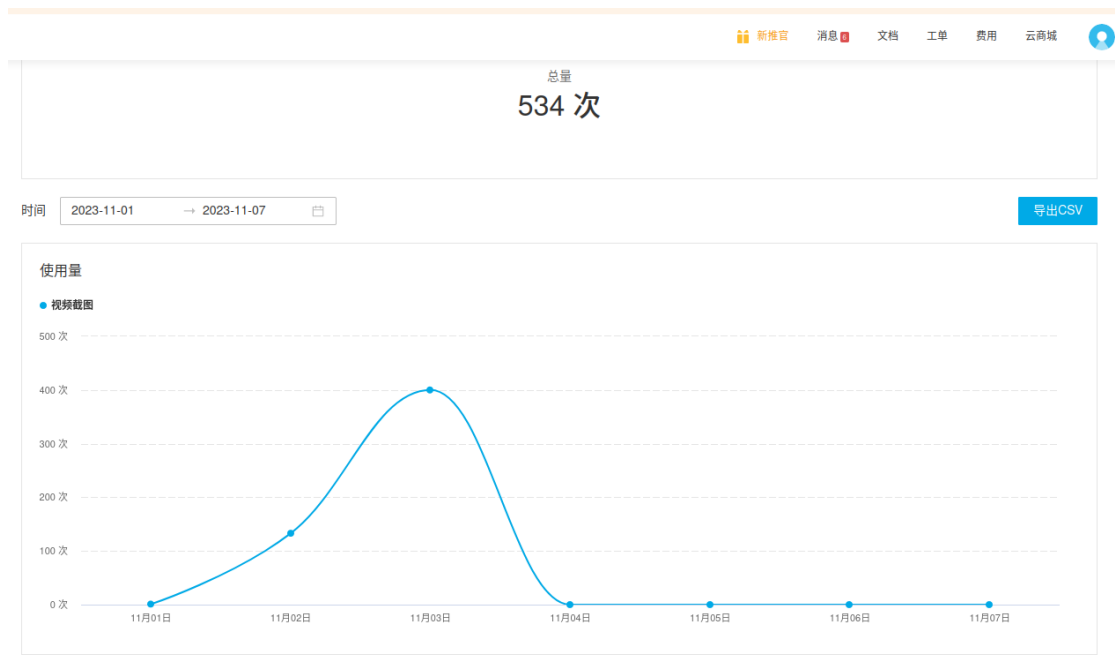
#### (4) Kodo:

使用七牛云对象存储功能，创建私有的存储空间。

在音视频处理应用方面，为存储空间添加截图预设，从而产生对应的视频封面，也统一存储到 Kodo 空间中，为了方便空间管理，为每个视频在上传至空间的时候，指定了全新的视频名称，并且伴随着对应的视频封面名称。

#### (5) 智能多媒体服务:

在音视频处理应用方面，为存储空间添加截图预设



每当有内容上传至对应的空间的时候，就使用视频截图功能，配置对应的触发器，将对应视频的封面产生并保存。

下面是对应的工作流配置



### 三、人员和分工

#### 队长-李海斌：

主要负责后端功能实现 + MySQL + Kodo 和视频素材收集以及画图。

#### 队员-马天瑞：

主要负责前端和视频素材收集以及文档的撰写。

#### 产品方案讨论：

产品方案由两人定期讨论来确定产品最终功能和效果。