\*红色标注的为需要改进的部分

**以下为网站的工作流程细分，以便大家再次复习一下网站的功能以及实现这些功能所需的步骤：**

**第一部分（用户输入）：**

首先用户会在前端的界面输入“公司名称”，这里会碰到三种情况，如下：

1. 该公司之前被搜过，然后是今天更新的资料库
   1. 最好的情况，资料库有，且资料最新最全（多久应该更新一遍资料库，怎么更新）
2. 该公司之前被搜过，但是是很久以前更新的资料库
   1. 第二好的情况，资料库有，但是资料不新
3. 该公司没有被搜过
   1. 不好的情况，所以需要解决
      1. 解决方案：如该公司没有被搜索，则让一个独立的服务器去完成整个搜索，筛选等流程，再提供给用户（这一步需要去考虑用时）

**第二部分（后台爬取）【写的可能不够细节，需补充】：**

用户输入完“公司名称”后，前端会到 database 找到匹配的资料然后返回给用户，那么这些 database 的资料是通过什么样的流程在一开始进入 database 的呢？

1. 爬虫会去已经爬好的网站（如发现报告）去进行文章爬取并加入 cache folder
   1. 在这一步会得到部分 parameters 如：文章标题，时间，页数，来源，种类（资讯/研报），URL
   2. 在后端爬文章的同时，也会有 whitelist 帮助其跳过已爬的文章
      1. 此处的 whitelist 就是根据文章的 url 或者文章在第三方网站的ID设定
      2. 想象一个场景：一个发现报告的文章，对比了美团和拼多多，然后用户第一次搜索美团，这个文章被爬下来且文章进入whitelist。但是如果用户搜索了拼多多，因为这个文章已经进入了whitelist，则会被跳过【如何解决】
         1. **解决方案：whitelist不仅是文章ID，也要同样的搜索关键词**
   3. 在后端爬文章的同时，可能会在某个网站的某个文章碰到问题，需要有机制跳过此文章。除此之外在运行时出现的任何问题都会被处理，并记录在error\_report文件里
   4. 备注：
      1. 对于199it 网站文章的爬取，公司名称只需出现一次就可以
         1. 解决方案：在爬取之前先看一下是不是199.it这个网站，这种情况最好少来，很烦！
      2. 对于html形式的文章，会自动转换html成text
2. 被加入 cache folder 的文章会被filter + process，然后得到所有其他的 parameters 如：文本，标签数量

**==== run\_filter() ====**

* 1. Filter：
     1. 将所有关键词出现次数不到3次（预设三次）的文章放入黑名单
  2. Process：
     1. 将pdf形式的文章转换成text并存储到json中
     2. 生成tags并存储到json中
     3. 应该将标题和标签数量都进行一个哈希值的计算，
        1. 如果被爬文章的标题一样，则归为同一文章
        2. 如果被爬文章标题不同，则计算标签哈希值。若标签哈希值一样或非常相似，则归为同一文章
     4. 在*每个网站*的folder里建立一个summary文件，过滤掉黑名单中的文章，将黑名单外的文章对应的json 整理后放进这个网站的summary文件中。等所有网站的文章都filter完，将所有*网站*的summary文件汇总成一个*Parent* *summary*文件。
  3. 此处可能会用到 OCR 功能，在OCR耗时时长这个问题没有解决的时候，先不要启用
  4. 此处提到的标签数量，***还没有将***标签放入一个大的分类，比如“股权架构”是属于“企业基本情况”这个大标签的（因为到时前端会需要点击“企业基本情况”来看到所有属于企业基本情况的标签）

1. 所有在 cache folder 的资料（*Parent summary*）一并更新到 database

**==== upload.update\_filtered()** **====**

* 1. 这个 database 是否已租

1. 最后清理 cache folder 准备下一次的文章爬取
   1. 这个 code 还需加工

【整个“第二部分”流程在后台无限循环】

**第三部分（信息呈现）：**

经过第一部分的企业名称输入，和第二部分的文章爬取以及 parameters 录入，database 可以反馈给用户符合要求且有质量的文章了，那么现在就要考虑文章的呈现，文章的呈现可分为以下的两个功能：

1. 排序
   1. 时间排序（需要把时间从早到晚的数值变成绝对值）
   2. 相关性排序（该数值暂定为公司名称出现次数）
      1. 这里之所以不用公司名称出现占比，是因为个别文章可能是很详细的行业调研，然后公司名称虽然出现占比会较少，但是该文章绝对有用
      2. 可用tf-idf算法计算关键词与文章的关联程度【与 Vincent 讨论】
   3. 综合排序（需要把时间从早到晚的数值变成绝对值 + 相关性数值）
2. 通过点击“标签”功能框，来精准看到拥有这些“标签”的文章
   1. 后端的 database 需要把标签的结构写清楚，然后约翰会联系外包的前端工程师来连接后端的 database 从而到达此功能（此处还需前端工程师介入后，才知道达到此功能的需求）

**第四部分：**

从一到三部分，网站的主要功能就已集齐，然后第四部分向包含一些一直以来说到的功能，此处参考整合sheet的“战略甘特图”：

1. 摘录
2. 预览
3. 在线笔记本
4. 公司名称自动联系公司业务（如果输入“谷歌”，系统可以自动得到谷歌的主营业务名称，比如说“广告”，然后产生新的标签，让用户在搜索“谷歌”后可以直观的看到“谷歌”的所有业务，以及所有拥有这些业务标签的文章）