1. 首页导航栏上给两个选择，一个按时间排布文章的，另一个是按关键字搜索。右上角放一个登录或注册的链接。

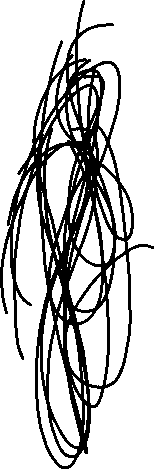
2. 文章发布、编辑页面。

3. 注册或登录页面。



4. 单个用户页面，这里如果可以的话，做个侧边栏，登录后在每个页面都可以展开的那种。

5. 单篇文章展示页面、评论点赞。



嗯…页面设计的话我还是会想就这么几个。编排的话从element ui上摘一摘稍微调一下问题应该不大。为什么我一开始花那么久倒弄那些html。。。。

具体实现上，可是……有很多有待说明的东西。

**第一个是跳转。**

首页我想用1当。登录注册链接3是放在角落里的。发布文章的页面2链接也可以放到首页的导航栏上。导航栏跳转也可以用路由，element-ui上就有说。

简略排布展示文章页1与详细展示页5的跳转。

以后可能到处都有的用户小logo与用户个人页面4的跳转。

这两个的实现可以用vue-router，跳转时顺便传递params参数，然后根据参数去向后台调取后一个页面需要的信息。问题是…不能刷新，一刷新那些参数就丢了，只能从各处点着进。可以的话，之后再看看有没有别的办法。——已解决，params可以通过：id（：参数）的方式写在路由路径上，不会影响页面加载。

**然后是权限、登录状态的问题。**

目前并不想写什么权限访问的东西，但发布东西的话…至少要有一个用户。

localStorage ，cookies ，sessions；还有一个vuex…可能要再学点东西了。个p哦，基本功能一个seesions就够了，只不过其他地方又有不少毛病。一开始是后台获取不到session里的登录信息，一直是null，然后有人说是前后端分离项目的跨域问题，然后在前台加了一个 this.axios.defaults.withCredentials = true;然后就继续报错，又去查，在后端那边又整了一个corsConfig，然后就成功了！

Vuex似乎是有着和后端三层架构类似的构造和功能…如果写分离的话可能不太需要。

**然后是文件上传问题。主要是图片和文章。**

我看到的是…文章直接变成text保存到数据库，图片文件很多并不放在数据库里，也许数据库里会放的是文件的地址链接。

还有一个是富文本编辑器的事，tinymce报错调了半天最后发现是版本不对才报错…

目前苛待解决的是后两个，压根还没开始。或者先把查询的写了。

还有，重写一下前台，太乱了。。

1. 重写前台，该展示的都弄出来，目前评论区输入框，登录页面，导航头都没弄好。
2. 文章按关键字查询
3. 用户登录
4. 图片，文章评论发表。

嗯。

————登录，算是弄好了，简陋但是还好。听说加上vuex可以整很多花活，里面也有类似后端那边的三层架构的东西，但还是算了。

————评论区都好了。评论区如果可以@人、可以整二级评论楼中楼就好了，但还是算了吧。

————图片上传没问题了，文章内的被tinymce转成base64直接传数据库里，其他的通过前后端已有的服务端口丢到本地文件夹。虽然专业似乎是会为文件专门整一个服务器，但这个就算了吧。

————查询只能按描述这一栏的数据查了吧…如果有高级的自动根据内容生成简略描述、查询的时候还自动匹配然后高亮展示文本就好了。但现在还是算了吧。

来一个二次畅想。细化、增改。

1. 首页导航栏上给两个选择，一个按时间排布文章的，另一个是按关键字搜索。右上角放一个登录或注册的链接。登录后此处变成一个导航栏4，可以打开的那种。文章展示方面，给两种选项，是块状还是条状。查询页面也想要点高级操作，比如多字段模糊查找，查找的时候按照匹配的字段高亮出对应文本，自动生成简略信息等等。

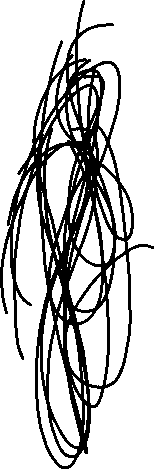
2. 文章发布、编辑页面。图片、文件可上传至第三方服务器。

3. 注册或登录页面。平平无奇。



4. 单个用户页面，这里如果可以的话，做个侧边栏，登录后在每个页面都可以展开的那种，以上已在1说明。具体详情页里、头像、介绍、发布过的文章；如果是登录用户本人的页面，用v-if控制显示出一些额外的后台模块、文章增删改、消息和@的提示记录......

5. 单篇文章展示页面、评论收藏点赞。评论区整出些花活，二级评论，互相@等等。



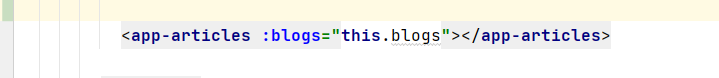
6. 收藏夹 favorites，tag这样的文章分类页面。

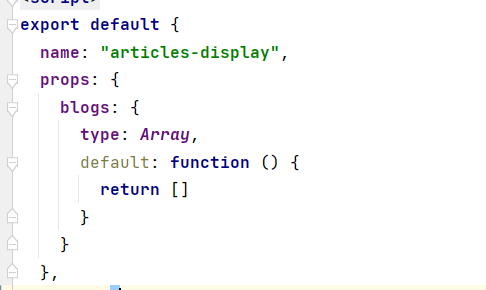
当然上面这样在结课前大概是不可能的。。。所以是想想。

说一下对vue的感觉。vue中提供了许多各种各样的数据绑定传输的小玩意，从{{ }}，v-model，v-bind对具体数据的绑定；到用来监听用户鼠标键盘上操作的v-on的click，keyup等等 这可能是最有趣的一部分了...因为要看的东西很短，有实时的交互，和css样式控制结合，很容易弄出些好玩的小玩具出来。

然后是，vueComponent里一些东西，name，methods，data，props，mounted……这些东西和java里的注释很像，但…不知道为什么vue里的看着就是舒服好多。

还有就是parent-child组件的问题，那个词用的很好，“注册组件”，而非面向对象里的“继承”，组件已经可以看做是一个对象了（虽然它还不是），而不是抽象的、未实现的类。这两个的parent-child根本不是一回事，在很多点上看起来相似又相反，所以对于传统的“父子”类这种叫法，我会想换一个，我会想把这里的parent-child叫成母子组件（其实我有过思考的！不是随便叫的！最简单的，“继承”体现的自顶向下，“注册”体现的是自底向上。）。





Props，作为子组件的那个组件里的props。那是给妈妈组件开一个接口，从妈妈那里拿数据传到子组件，经过子组件处理后展示在子组件的template里，然后再由妈妈组件调用、一整个展示。（为什么有种奇怪的感觉）（路由的props可以和组件的交互！组件还可以和路由xxoo（不是））

$refs，在元素标签，在vue自定义的小子组件标签里可以写上一个ref=“xxx”，然后就可以在妈妈组件里用this.$refs.xxx拿到那个元素，或者那一整个child-VueComponent，子组件。还有就是在vue里用this，指到的就是那一个vue组件…？

组件间通信还可以通过一个叫“全局事件总线”的东西，类似任意门那种，不用妈妈孩子一层层的传，可以跳。但具体我不了解。

项目启动，vue这边是maim.js引入App.vue，把它作为一个子组件，然后，创建一个妈妈的妈妈？——#app，将App作为子组件；而之后的组件一般都作为App.vue的子组件，或者直接在router的index.js里import一下，直接跳转就好。

main.js里有可以导入插件，然后vue.use(xxx)一下就能全局使用。Vue，router，element-ui，axios基本都是全局导入的。

App.vue……有点被冷落的说。我就在根目录上写了一个goto 的按钮，然后就是router-view了，我登录页面弄了两层router-view嘿嘿。。还发现子路由也要写全、或者叫绝对路径…我不知道./这样的写法能不能用，但是不像那些有个baseURL可以直接拼接。

里面似乎还有个叫生命周期的东西，控制页面渲染？methods里的函数通常要通过用户的事件触发，而生命周期钩子mounted、created等等是根据vue的生命周期过程进行触发？

对，这个地方就是根据vue的进程自动触发一些东西，比如向后台发数据用来填充页面。不用用户操作自动挂载。

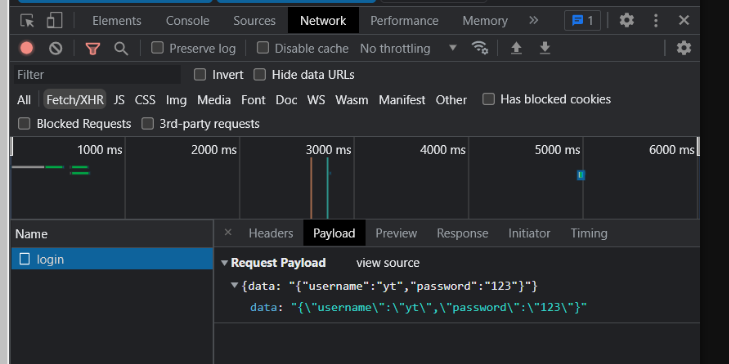
以及，对写这些玩意整体的感觉。

连接是vue通过axios向后端发请求，然后靠java应用调数据库的数据；调出来的数据一般是json形式，通过key-value对应能贴到表格或其他什么地方或者进行些别的什么操作…。

好像没有太多神奇的地方，但要让整个过程顺利的进行的确要废不少功夫。

Vue还比较方便的地方可能是叫数据代理，想要拿到哪个直接this.data.xxx就行。母女组件通信什么的，..我没怎么用，毕竟写的东西也不多。

对了还有一件事，从后台传数据到前台能先直接console.log一下看着就用，但前台到后台，传数据到后台就…要用不一样的方式去看，找开发者工具的network 看XHR这种请求下的payload，这里会有发送出去的数据。



axios.post（url{data}）这样简写发送数据总是会带上一层data:{}这玩意，然后笨蛋后台就解不了了。这玩意算是坑了我好久。。。。其实早就想过这种可能，毕竟后台传来数据都是要先console.log一下再搞，很多时候都tm有好几层data，用的时候得写成this.data.data.data.xxx。。。但苦于不知道怎么看post出去的数据到底长啥样。一看就明白了。