

项目说明

框架

使用 create-react-app 脚手架创建的 React（类似 Vue）项目。

主要用到了 Material UI 来美化。 <https://mui.com/material-ui/>

运行

- 需要有node.js环境。

clone后，运行命令：

```
npm install
```

```
npm start
```

访问 <http://localhost:3000/> 就可以看到

项目结构

```
.
|-- README.md
|-- package-lock.json
|-- package.json
|-- public
|   |-- index.html
|   |-- logo.png
|   |-- manifest.json
|   `-- robots.txt
|-- src
|   |-- App.js                // 主组件
|   |-- assets
|   |   |-- fonts
|   |   |   |-- SourceHanSansSC-Regular.otf
|   |-- components           // 存放React组件，用于实现页面的部分功能
|   |   |-- DatasetsView.js
|   |   |-- PapersView.js
|   |   |-- ReportSection.js
|   |   |-- Steps.js
|   |   |-- StyledAppBar.js
|   |-- index.css
|   |-- index.js             // 入口js文件
|   |-- pages                // 存放各个页面文件 用于路由
|   |   |-- HomePage.js
|   |   |-- ManageProject.js
|   |   |-- NewProject.js
|   |   |-- Test.js
|   |   |-- steps
|   |       |-- DryExperiments.js
|   |       |-- ExperimentalDesign.js
|   |       |-- LiteratureProcessing.js
```

```

| | | | -- Programming.js
| | | | -- Search.js
| | | | -- utils
| | | | -- DatasetsProcess.js
| | | | -- PapersetProcess.js

```

`index.js` 是整个项目的入口文件，在 `index.js` 中引入主组件 `App.js`

`App.js` :

```

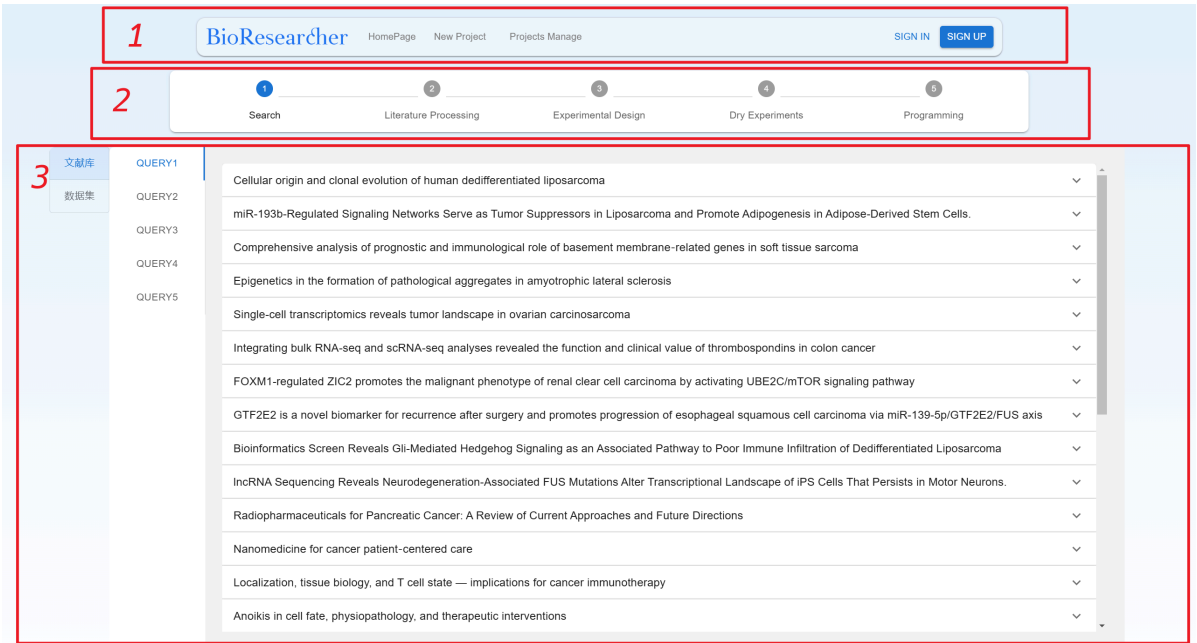
function App() {
  return (
    <div className="App">
      <AppAppBar />
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/new-project" element={<NewProject />} />
        <Route path="/manage-project" element={<ManageProject />} />
        <Route path=":id/steps" element={<Steps />}>
          <Route index element={<Navigate to="search" />} />
          <Route path="search" element={<Search />} />
          <Route path="literature-processing" element={<LiteratureProcessing />} />
        </Route>
        <Route path="experimental-design" element={<ExperimentalDesign />} />
        <Route path="dry-experiments" element={<DryExperiments />} />
        <Route path="programming" element={<Programming />} />
      </Routes>
      <Route path="/test" element={<Test />} />
    </div>
  );
}

export default App;

```

根据不同的url（代码中的path参数），路由到不同的页面

举例：



1 对应： `components/StyledAppBar.js`

2 对应： `components/Steps.js`

3 对应： `pages/steps/Search.js`

其中 3 主要分为两个部分：左侧的两个按钮（文献库和数据集），右侧的数据展示部分。

因为要显示分别文献库和数据库，我写了两个组件来实现，对应 `components/Papersview.js` 和 `components/DatasetsView.js`

在 `search.js` 中有段代码：会判断显示哪一个

```
const renderContent = () => {
  switch (view) {
    case 'papers':
      return <PapersView />;
    case 'datasets':
      return <DatasetsView />;
    default:
      return null;
  }
};
```

`Paperview.js` 中用到了 Material UI 的 tab 组件，左侧标签栏，右侧内容。

共享文档里的页面要求：

基于大模型的生物医学自动科研系统

前端

一、登录页

二、首页：

1. 创建项目（项目名，描述，...）
2. 选择项目，编辑项目，删除项目，...

三、项目页：

1. 输入框（实验目的、实验条件、实验要求）
2. 检索数据库（待实现）：文献库（PMC, PubMed, ...），数据集检索库（GEO, NCBI, cBioPortal）
3. 用户自主选择要上传的参考文献
4. 模型选择（GPT4o, gemini, claude, ...）
5. 其他参数（，超参数，时间与质量的权衡，比如参考文献的数量、审稿人评估最大轮次等）
6. 提交/开始按钮

四、Search 子页：

1. 按文献和数据集分成两部分，生成的 query 作为标签栏标题，点击展开对应的检索内容（溯源）：query 有两部分，用于检索paper的和用于检索dataset的
2. 检索文献与数据集的框内显示过滤器评估的可用性分数，点击可用性分数显示模型评估理由（叹号提示）。（这里的理由先留个位置就可以，可以先用其他字符串代替；）
3. 检索内容的统计结果展示，如检索到文献及可用文献柱状图、数据集柱状图等（待完善）
4. 筛出的、成功下载下来的文献或数据集分别高亮表示+标注
5. 检索到的数据集与相关描述信息：dataset/retrieved_datasets_info.json
6. 过滤后的数据集与相关描述信息：dataset\related_datasets_info.json
7. 检索到的paper：papers\related_papers_info.json
8. 过滤后的paper：papers\retrieved_papers_info.json
9. papers下的pdf是成功下载的paper，但可能并不是related_papers_info.json 里的所有，因为有一些爬不下来。所有related_papers_info.json里的论文都给一个小框，点击展示title 和摘要。没成功下载的就标注一下，比如在标题后面加一个小括号 "(需手动下载)"

五、Literature Processing 子页：

1. 每个文献作为一个标签栏，点击显示处理详情子页面
2. 在处理详情子页面显示生成的报告以及分析结果，每块内容（即一级标题、大纲的小标题，步骤内容）均可点击，点击显示来源、模型生成的推理原因、审稿人评审修改过程，三者以按钮的形式再嵌套为“点击展开”
3. 每篇paper会有一个对应的实验报告文件和分析文件，把papers里的 ".pdf" 换成 "report.json" 和 "analysis.json" 就是这两个文件了
4. 实验报告文件有很多 part，每个 part 有多个step，step 里面的implementation details 就是本步骤的具体内容，results 是实验结果，results original text 是实验结果对应到原文中的内容，这些需要在实验报告的详情子页面中显示

5. 分析文件会对实验报告中的每个part进行分析，需要把分析的这些（参考性，原因，建议）也显示在详情子页面对应的part部分

六、Experimental Design 子页（关键页面）：

1. 与 Literature Processing 类似，点击显示详情
2. 交互栏（待实现）：用户输入它的修改建议，提交后返回修改结果，并将修改部分高亮
3. 这个页面需要展示最终生成的实验方案，对应论文中的experimental protocol，就是 experiment_program1.json 这三个文件，现在的设定是对每个用户输入的实验目的会生成三版，我们先把这三版都展示一下

七、Dry Experiments 子页：

1. 逐层展示提取出的干实验，点击显示详情
2. 对应dry_experiment1.json 三个文件

八、Programming 子页：

1. 以提取出的干实验的格式逐层按标签栏展示代码，以及代码运行结果和 log
2. 代码执行模块调试

进阶要求：

动态演示生成过程？

流式生成？