Project Document for CrazyPages

Author: Zheng Hanwen

StudentID: 1853931

Tutor: Liu Yan

1. Abstraction

This document is a project document for the assigned individual project for Web Service & SOA course. It's written as an entertainment auxiliary for every single person, especially who surf the Internet a lot. For some objective reasons, the theme of my project has changed after the project proposal submitted. This document aims to draw a brief illustration of CrazyPages and make it easier to learn about its technology stack. This proposal is divided into 5 parts mainly. The first part intoduces the design concept and practicality of CrazyPages, in addition to some operation guides. Then the second part demonstrates the architecture of the project, divided by the front-end and the back-end design. The third lists all the web-apis I've used, demonstrates all the apis that I design and achieve, and explains the integration. The fourth part demonstrates a few screenshots of my project. The last part instructs readers to set up environment and precondition in order to run the whole project.

2. Brief Introduction

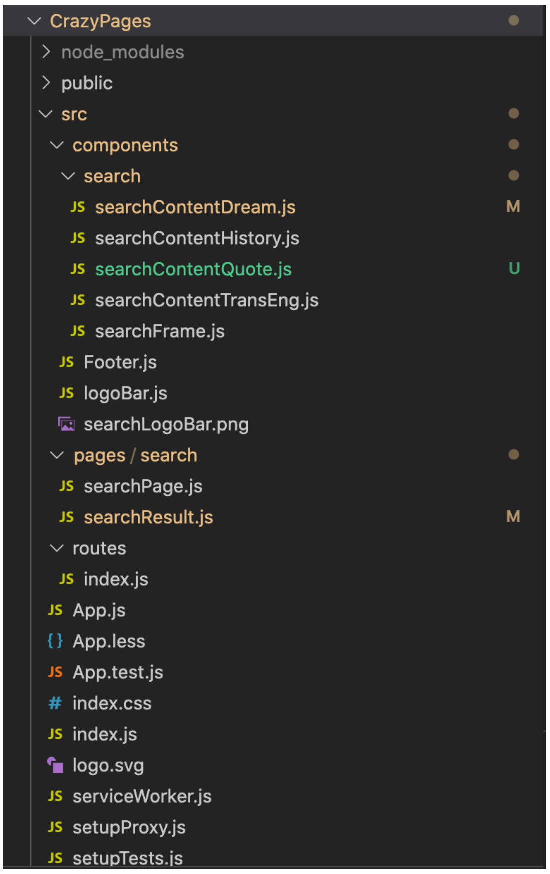
CrazyPages is a product made by a crazy programmer(yey, that's me), to provide precious entertainment for high-speed living people and high-pressurers. I try my best to make the front-end pages cute and active. They may make you smile and also learn some interesting facts. Hope my project bring you a good day!

3. Architecture

3.1 Front-end

3.1.1 Introduction

My front-end part is based on a react scaffolding. React basically helps to distribute task into lower levels and realize easier data fetching and access. The structure is listed as below.



```
CrazyPages
  node_modules
  public
  src
    components
      search
        searchContentDream.js        M
        searchContentHistory.js
        searchContentQuote.js        U
        searchContentTransEng.js
        searchFrame.js
      Footer.js
      logoBar.js
      searchLogoBar.png
    pages / search
      searchPage.js
      searchResult.js                M
    routes
      index.js
    App.js
    App.less
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
    setupProxy.js
    setupTests.js
```

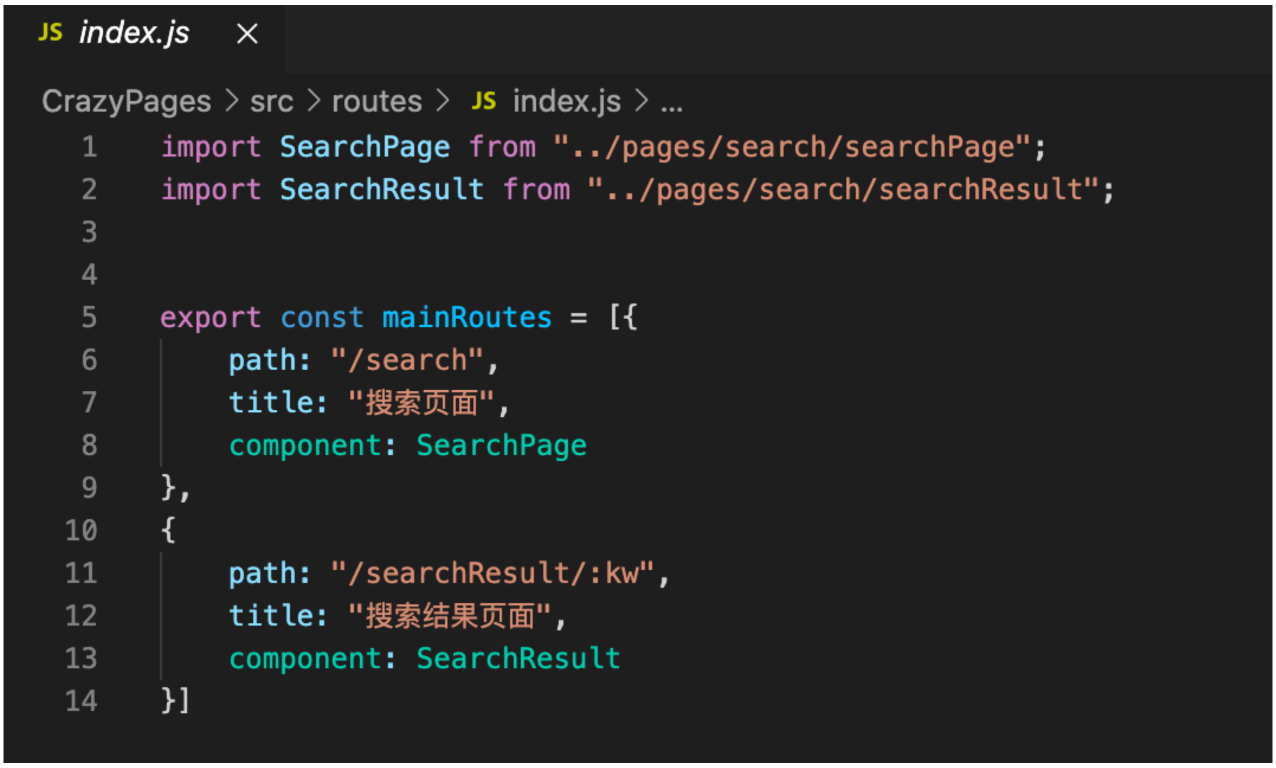
3.1.2 Structure

- node_modules: dependency file, fine to be added into git.ignore
- public: openly access files
- src: edit your 90% codes here
- index.js => load some global scripts and start a react app
- **component** => including views and logical controller of the elements of frontend web pages
- pages: including more direct rendering of pages
- **routes** => **index.js** ==> package React routes to jump to a certain path and define mine
- craco.config.js => realize the global configuration of UI component library such as change theme color
- setupProxy.js => import http-proxy-middleware and set up my Proxy agent

3.1.3 Most Used Libraries

1. ant-design UI
2. axios
3. proxy
4. jquery

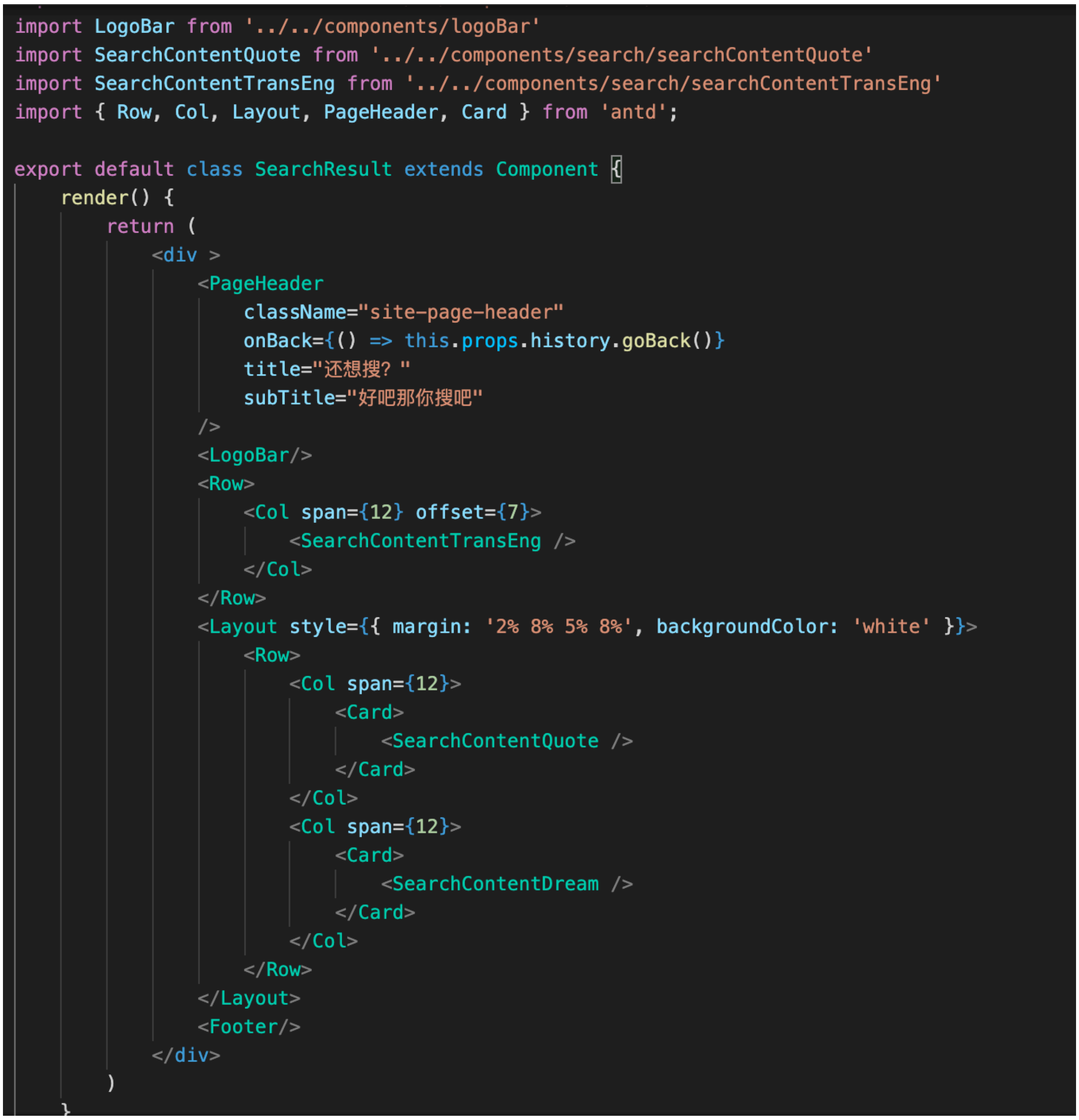
3.1.4 Essential Code Pics


JS index.js ×

CrazyPages > src > routes > JS index.js > ...

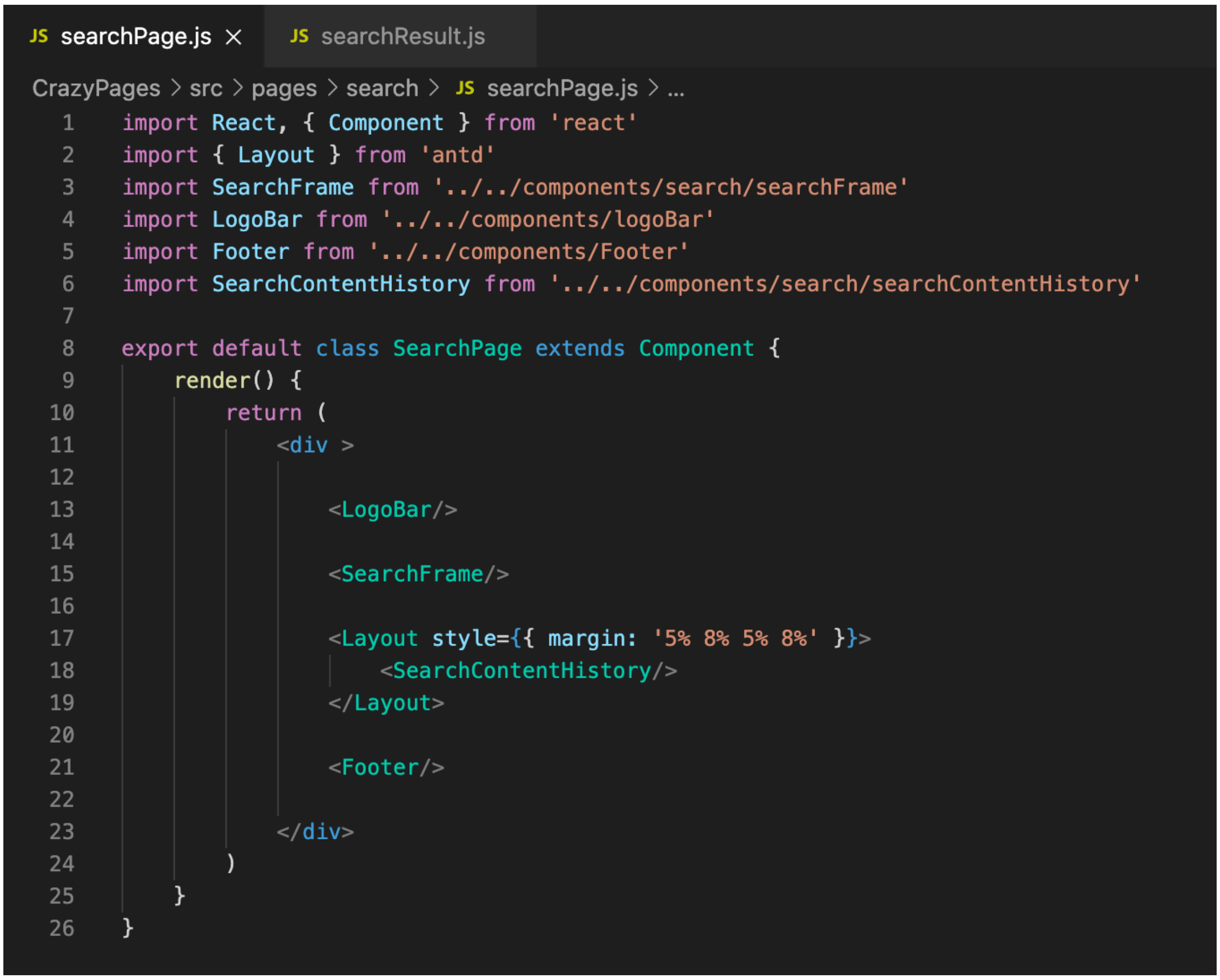
```
import SearchPage from "../pages/search/searchPage";
import SearchResult from "../pages/search/searchResult";


export const mainRoutes = [{
    path: "/search",
    title: "搜索页面",
    component: SearchPage
},
{
    path: "/searchResult/:kw",
    title: "搜索结果页面",
    component: SearchResult
}]
```

pic1: set routes



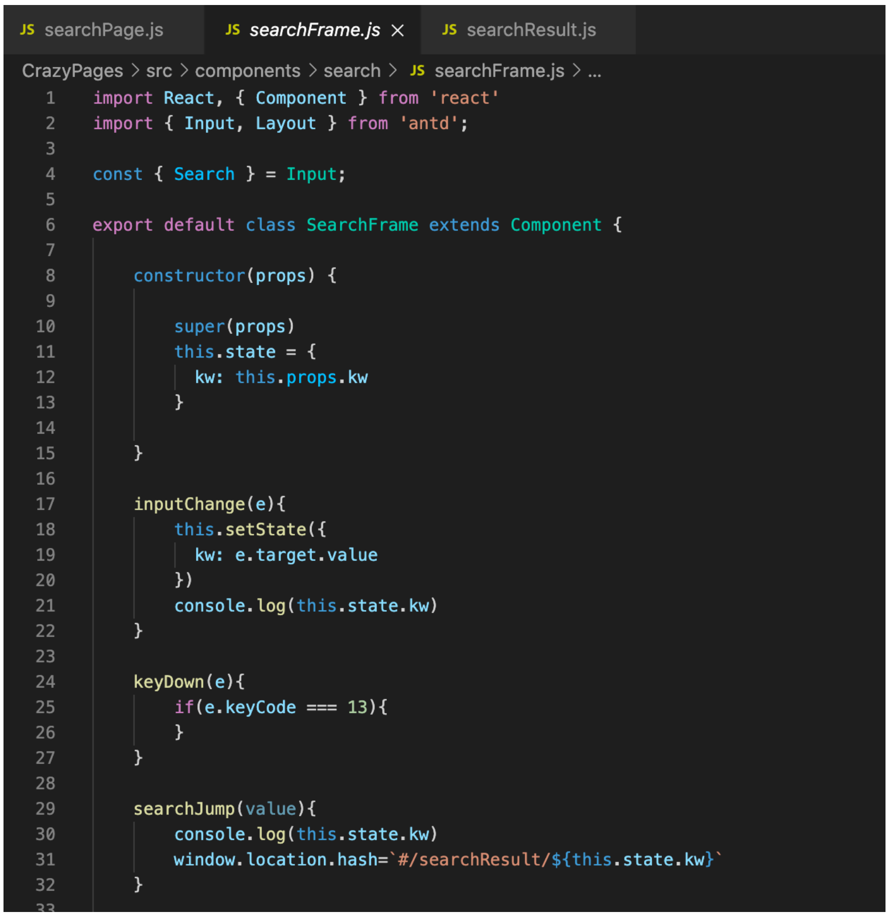
```
import LogoBar from '../../components/logoBar'
import SearchContentQuote from '../../components/search/searchContentQuote'
import SearchContentTransEng from '../../components/search/searchContentTransEng'
import { Row, Col, Layout, PageHeader, Card } from 'antd';

export default class SearchResult extends Component {
    render() {
        return (
            <div >
                <PageHeader
                    className="site-page-header"
                    onBack={() => this.props.history.goBack()}
                    title="还想搜? "
                    subTitle="好吧那你搜吧"
                />
                <LogoBar/>
                <Row>
                    <Col span={12} offset={7}>
                        <SearchContentTransEng />
                    </Col>
                </Row>
                <Layout style={{ margin: '2% 8% 5% 8%', backgroundColor: 'white' }}>
                    <Row>
                        <Col span={12}>
                            <Card>
                                <SearchContentQuote />
                            </Card>
                        </Col>
                        <Col span={12}>
                            <Card>
                                <SearchContentDream />
                            </Card>
                        </Col>
                    </Row>
                </Layout>
                <Footer/>
            </div>
        )
    }
```

pic2: SearchResult Component design



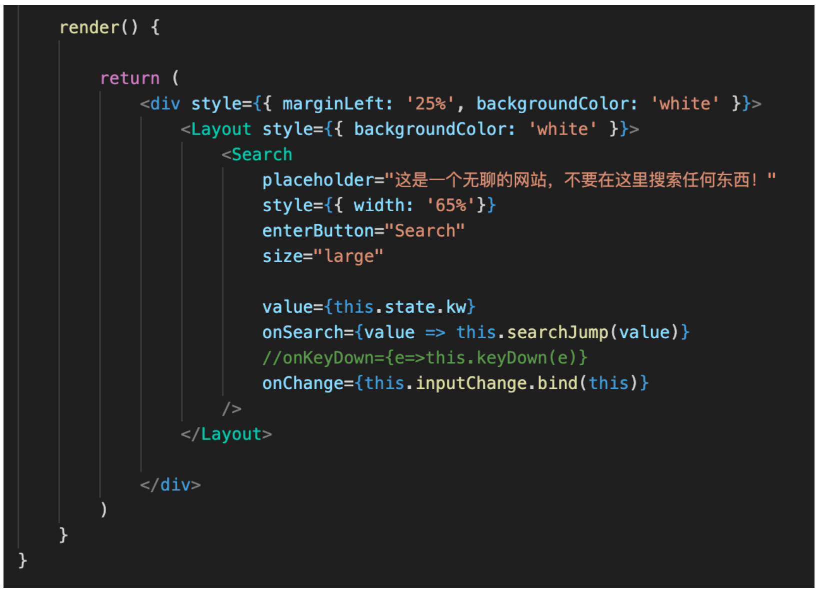
```
import React, { Component } from 'react'
import { Layout } from 'antd'
import SearchFrame from '../../components/search/searchFrame'
import LogoBar from '../../components/logoBar'
import Footer from '../../components/Footer'
import SearchContentHistory from '../../components/search/searchContentHistory'

export default class SearchPage extends Component {
    render() {
        return (
            <div >

                <LogoBar/>

                <SearchFrame/>

                <Layout style={{ margin: '5% 8% 5% 8%' }}>
                    <SearchContentHistory/>
                </Layout>

                <Footer/>

            </div>
        )
    }
}
```

pic3: SearchPage Component design



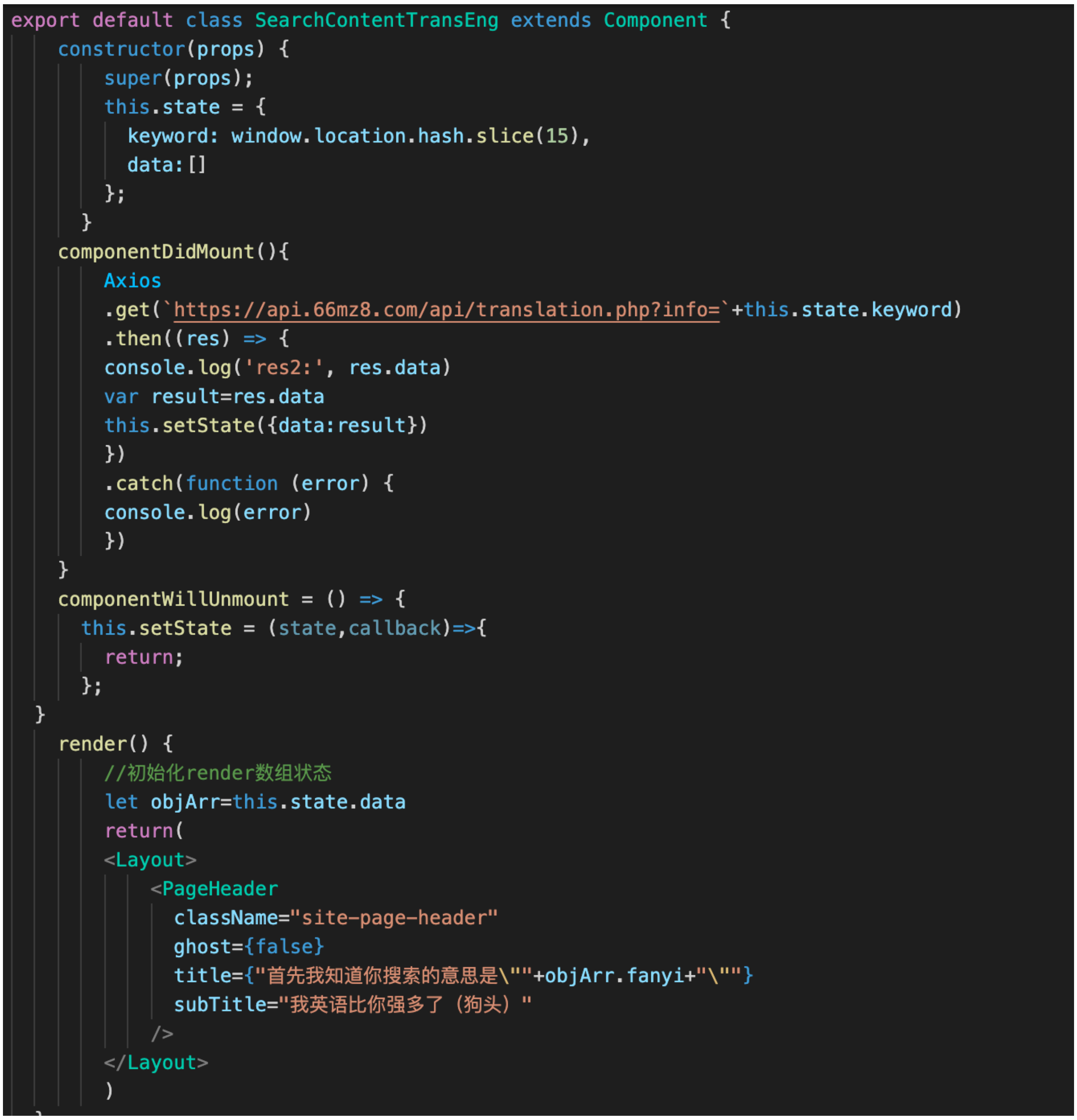
JS searchPage.js
JS searchFrame.js
JS searchResult.js
CrazyPages > src > components > search > JS searchFrame.js > ...

```
import React, { Component } from 'react'
import { Input, Layout } from 'antd';

const { Search } = Input;

export default class SearchFrame extends Component {

    constructor(props) {

        super(props)
        this.state = {
          kw: this.props.kw
        }

    }

    inputChange(e){
        this.setState({
          kw: e.target.value
        })
        console.log(this.state.kw)
    }

    keyDown(e){
        if(e.keyCode === 13){
        }
    }

    searchJump(value){
        console.log(this.state.kw)
        window.location.hash=`#/searchResult/${this.state.kw}`
    }
```



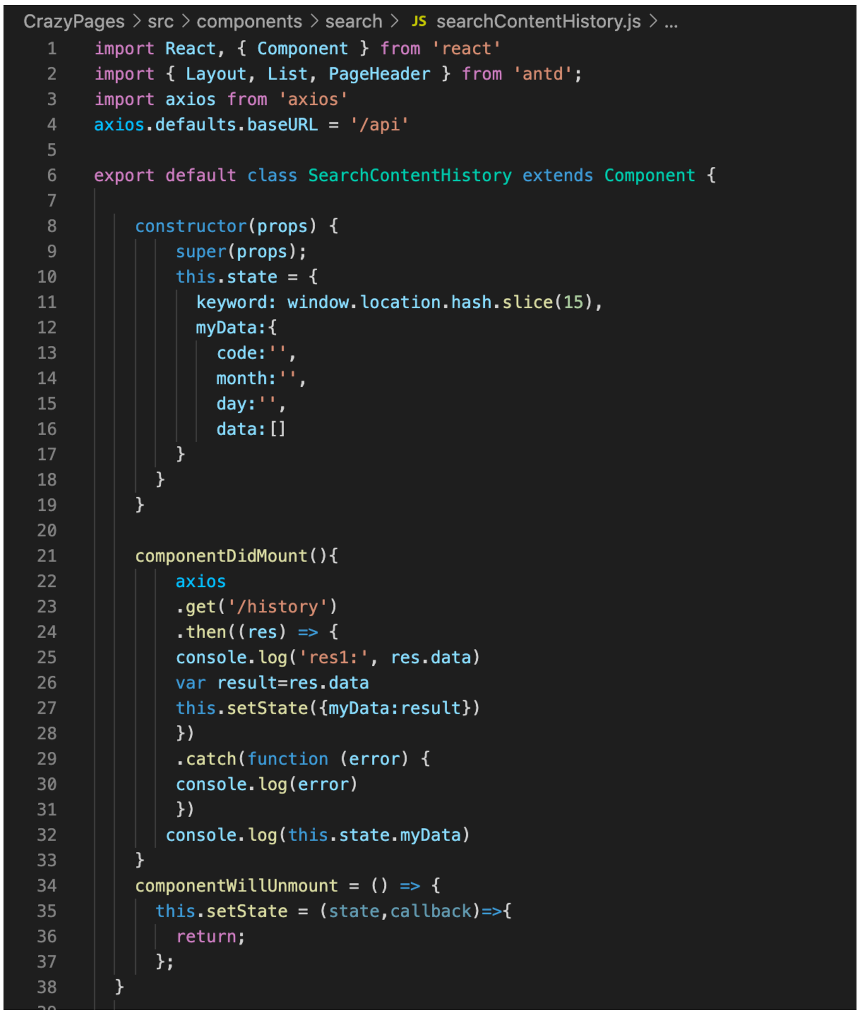
```
    render() {

        return (
            <div style={{ marginLeft: '25%', backgroundColor: 'white' }}>
                <Layout style={{ backgroundColor: 'white' }}>
                    <Search
                        placeholder="这是一个无聊的网站，不要在这里搜索任何东西！"
                        style={{ width: '65%'}}
                        enterButton="Search"
                        size="large"

                        value={this.state.kw}
                        onSearch={value => this.searchJump(value)}
                        //onKeyDown={e=>this.keyDown(e)}
                        onChange={this.inputChange.bind(this)}
                    />
                </Layout>

            </div>
        )
    }
}
```

pic4-5: SearchFrame Component design



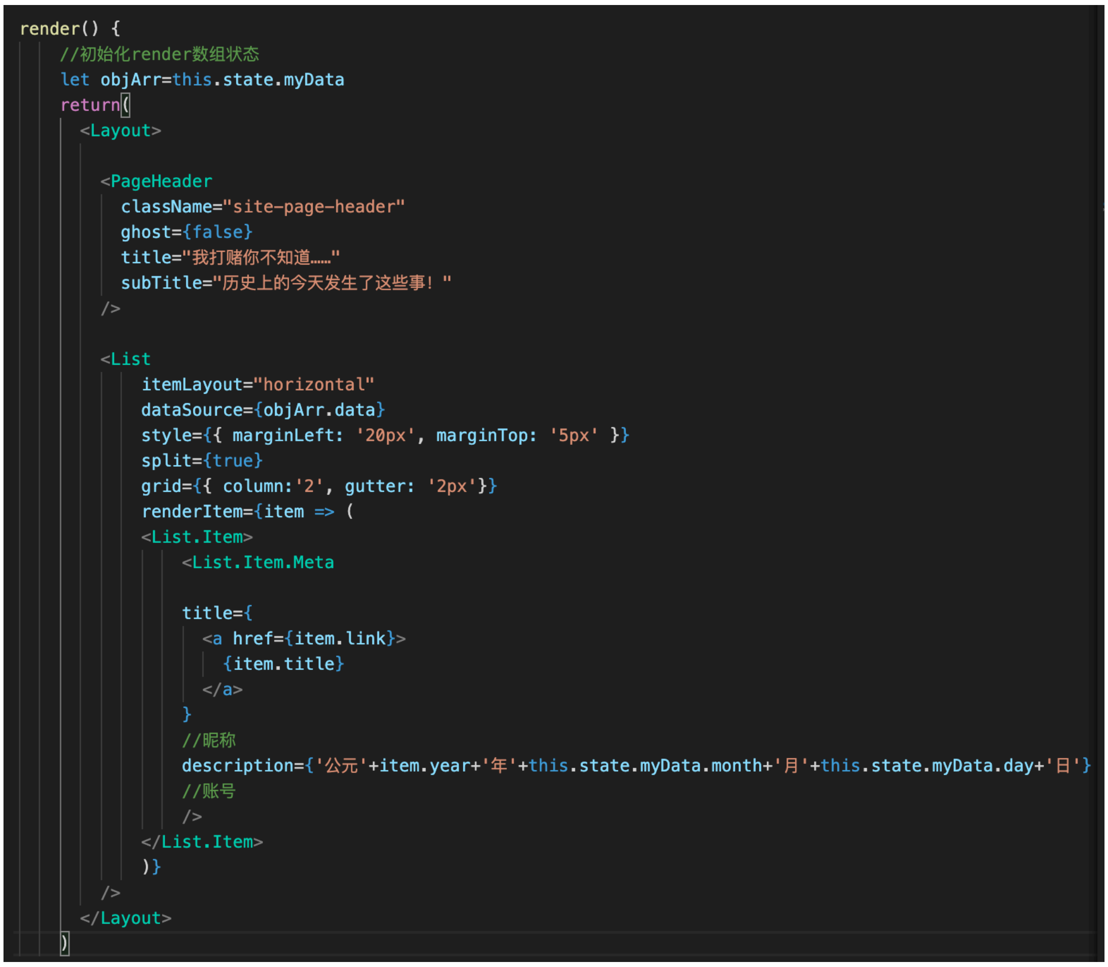
```
export default class SearchContentTransEng extends Component {
  constructor(props) {
    super(props);
    this.state = {
      keyword: window.location.hash.slice(15),
      data:[]
    };
  }
  componentDidMount(){
    Axios
    .get(`https://api.66mz8.com/api/translation.php?info=`+this.state.keyword)
    .then((res) => {
    console.log('res2:', res.data)
    var result=res.data
    this.setState({data:result})
    })
    .catch(function (error) {
    console.log(error)
    })
  }
  componentWillUnmount = () => {
    this.setState = (state,callback)=>{
      return;
    };
 }
  render() {
    //初始化render数组状态
    let objArr=this.state.data
    return(
    <Layout>
      <PageHeader
        className="site-page-header"
        ghost={false}
        title={"首先我知道你搜索的意思是\""+objArr.fanyi+"\""}
        subTitle="我英语比你强多了（狗头）"
      />
    </Layout>
    )
```

pic6: SearchContentTransEng Component design, including fetch data through web-api by the way of URL



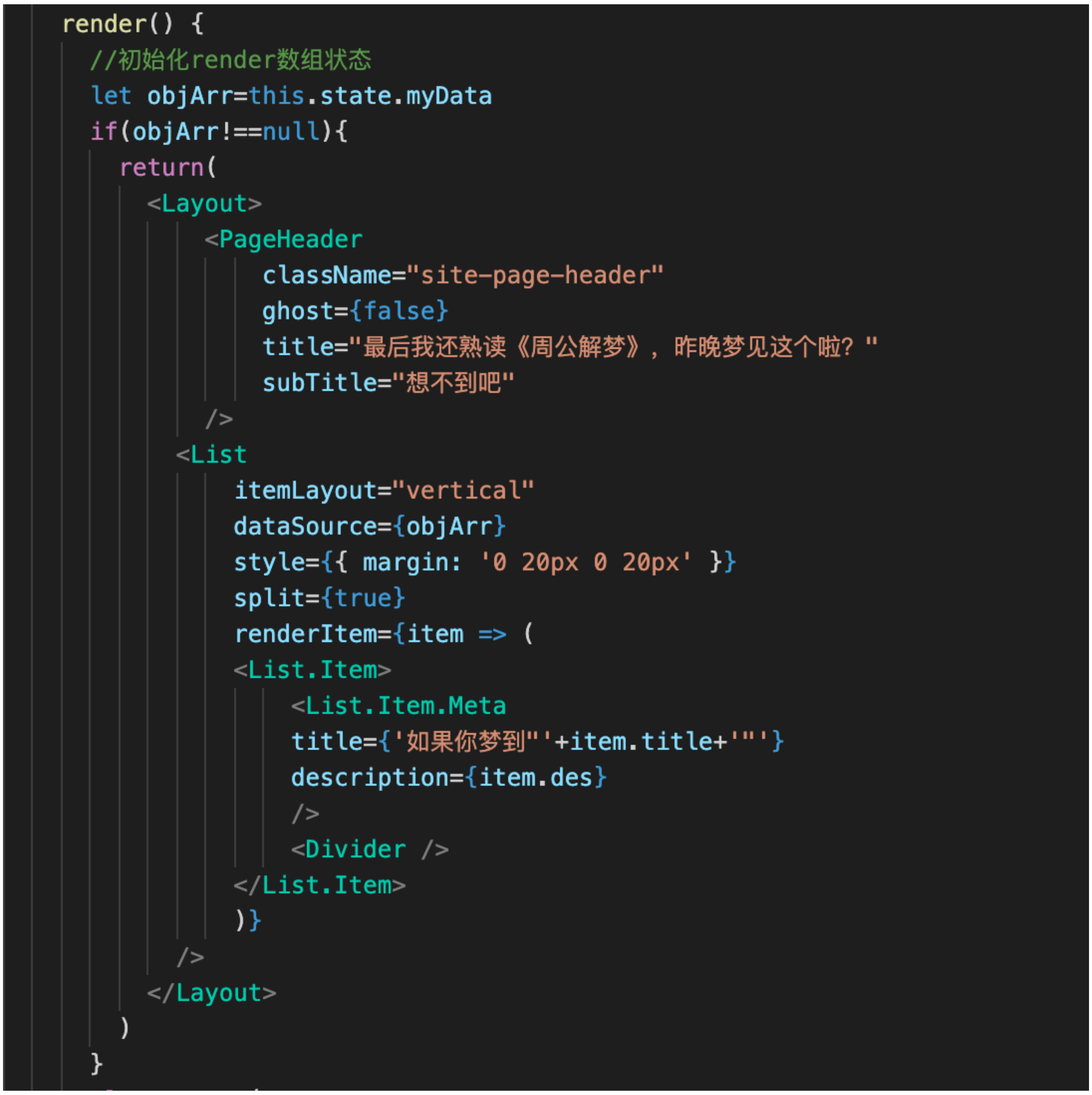
CrazyPages > src > components > search > JS searchContentHistory.js > ...

```
import React, { Component } from 'react'
import { Layout, List, PageHeader } from 'antd';
import axios from 'axios'
axios.defaults.baseURL = '/api'

export default class SearchContentHistory extends Component {

    constructor(props) {
        super(props);
        this.state = {
          keyword: window.location.hash.slice(15),
          myData:{
            code:'',
            month:'',
            day:'',
            data:[]
          }
        }
    }

    componentDidMount(){
        axios
        .get('/history')
        .then((res) => {
        console.log('res1:', res.data)
        var result=res.data
        this.setState({myData:result})
        })
        .catch(function (error) {
        console.log(error)
        })
       console.log(this.state.myData)
    }
    componentWillUnmount = () => {
      this.setState = (state,callback)=>{
        return;
      };
  }
```

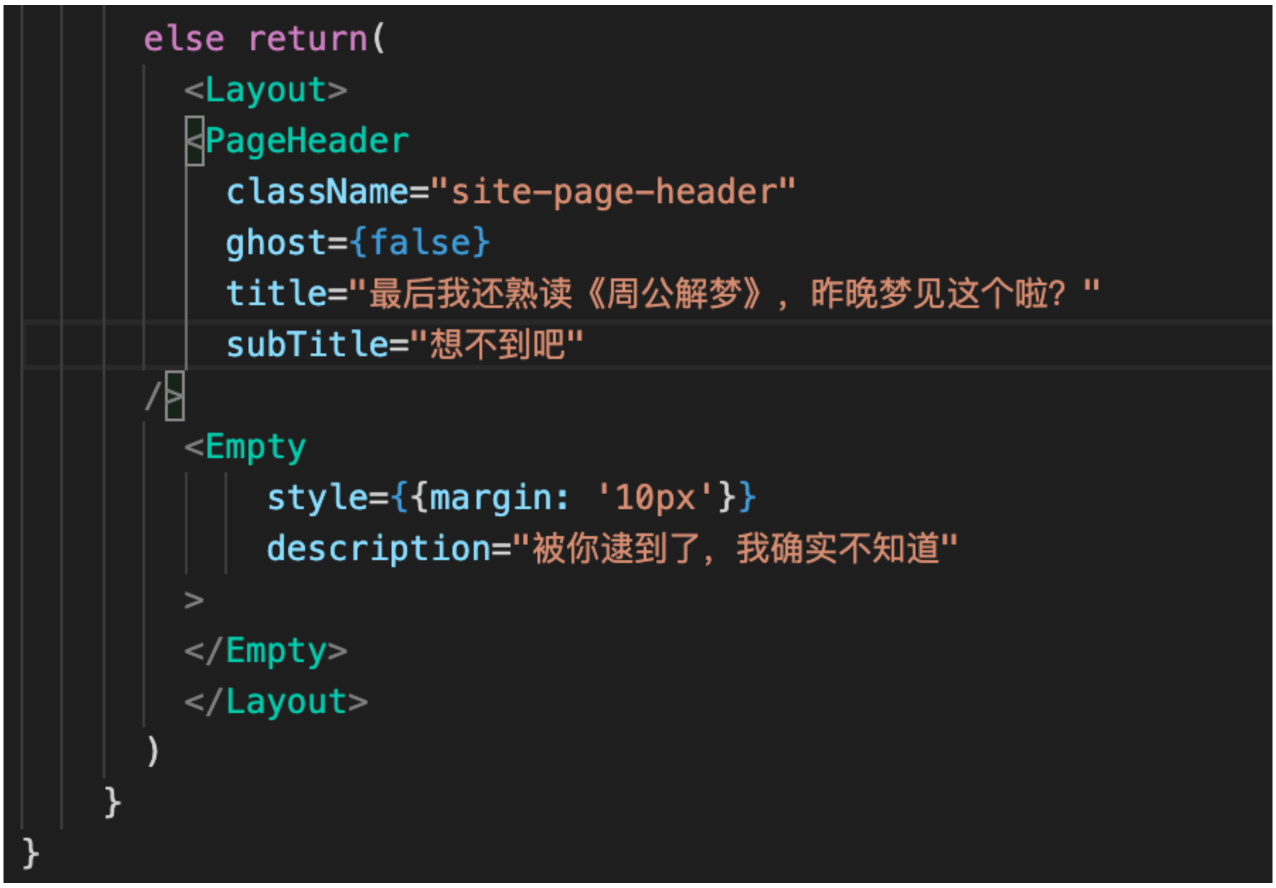


```
render() {
    //初始化render数组状态
    let objArr=this.state.myData
    return(
      <Layout>

        <PageHeader
          className="site-page-header"
          ghost={false}
          title="我打赌你不知道……"
          subTitle="历史上的今天发生了这些事！"
        />

        <List
            itemLayout="horizontal"
            dataSource={objArr.data}
            style={{ marginLeft: '20px', marginTop: '5px' }}
            split={true}
            grid={{ column:'2', gutter: '2px'}}
            renderItem={item => (
            <List.Item>
                <List.Item.Meta

                title={
                  <a href={item.link}>
                    {item.title}
                  </a>
                }
                //昵称
                description={'公元'+item.year+'年'+this.state.myData.month+'月'+this.state.myData.day+'日'}
                //账号
                />
            </List.Item>
            )}
        />
      </Layout>
    )
```

pic7-8: SearchContentHistory Component design, including fetch data through my api by agent name



```
render() {
  //初始化render数组状态
  let objArr=this.state.myData
  if(objArr!==null){
    return(
      <Layout>
        <PageHeader
          className="site-page-header"
          ghost={false}
          title="最后我还熟读《周公解梦》，昨晚梦见这个啦？"
          subTitle="想不到吧"
        />
      <List
        itemLayout="vertical"
        dataSource={objArr}
        style={{ margin: '0 20px 0 20px' }}
        split={true}
        renderItem={item => (
        <List.Item>
          <List.Item.Meta
          title={'如果你梦到"'+item.title+'"'}
          description={item.des}
          />
          <Divider />
        </List.Item>
        )}
      />
      </Layout>
    )
  }
```



```
            else return(
              <Layout>
              <PageHeader
                className="site-page-header"
                ghost={false}
                title="最后我还熟读《周公解梦》，昨晚梦见这个啦？"
                subTitle="想不到吧"
            />
              <Empty
                  style={{margin: '10px'}}
                  description="被你逮到了，我确实不知道"
              >
              </Empty>
              </Layout>
            )
        }
}
```

pic9-10: SearchContentDream Component design, featuring fault-tolerance assurance with renderung empty status

Below is my XmlToJson function:

```
// 调用函数将XML转换为JSON
function XmlToJson() {
}
XmlToJson.prototype.setXml = function(xml) {
    if(xml && typeof xml == "string") {
        this.xml = document.createElement("div");
        this.xml.innerHTML = xml;
        this.xml = this.xml.getElementsByTagName("*")[0];
    }
    else if(typeof xml == "object"){
        this.xml = xml;
    }
};
XmlToJson.prototype.getXml = function() {
    return this.xml;
};
XmlToJson.prototype.parse = function(xml) {
    this.setXml(xml);
    return this.convert(this.xml);
};
```

```
XmlToJson.prototype.convert = function(xml) {
    if (xml.nodeType !== 1) {
        return null;
    }
    var obj = {};
    obj.xtype = xml.nodeName.toLowerCase();
    var nodeValue = (xml.textContent || "").replace(/(\r|\n)/g,
"").replace(/^\s+|\s+$/g, "");
    if(nodeValue && xml.childNodes.length === 1) {
      obj.text = nodeValue;
    }
    if (xml.attributes.length > 0) {
        for (var j = 0; j < xml.attributes.length; j++) {
            var attribute = xml.attributes.item(j);
            obj[attribute.nodeName] = attribute.nodeValue;
        }
    }
    if (xml.childNodes.length > 0) {
        var items = [];
        for(var i = 0; i < xml.childNodes.length; i++) {
            var node = xml.childNodes.item(i);
            var item = this.convert(node);
            if(item) {
                items.push(item);
            }
        }
        if(items.length > 0) {
            obj.items = items;
        }
    }
    return obj;
};
```

3.2 Back-end

3.2.1 Introduction

My back-end part is based on Python Flask. I've learned how to use Flask to finish enough requirements for my project just for one night and finally realize everything necessary. Python Flask is agile, efficient and practical. And it's so suitable for this assignment. The structure is listed as below.

3.2.2 Structure

in main.py, you will first find dependencies imported such as flask, flask_cors (to solve CORS problems), requests.

```
# r'/*' 是通配符，让本服务器所有的URL 都允许跨域请求
CORS(app, resources=r'/*')
```

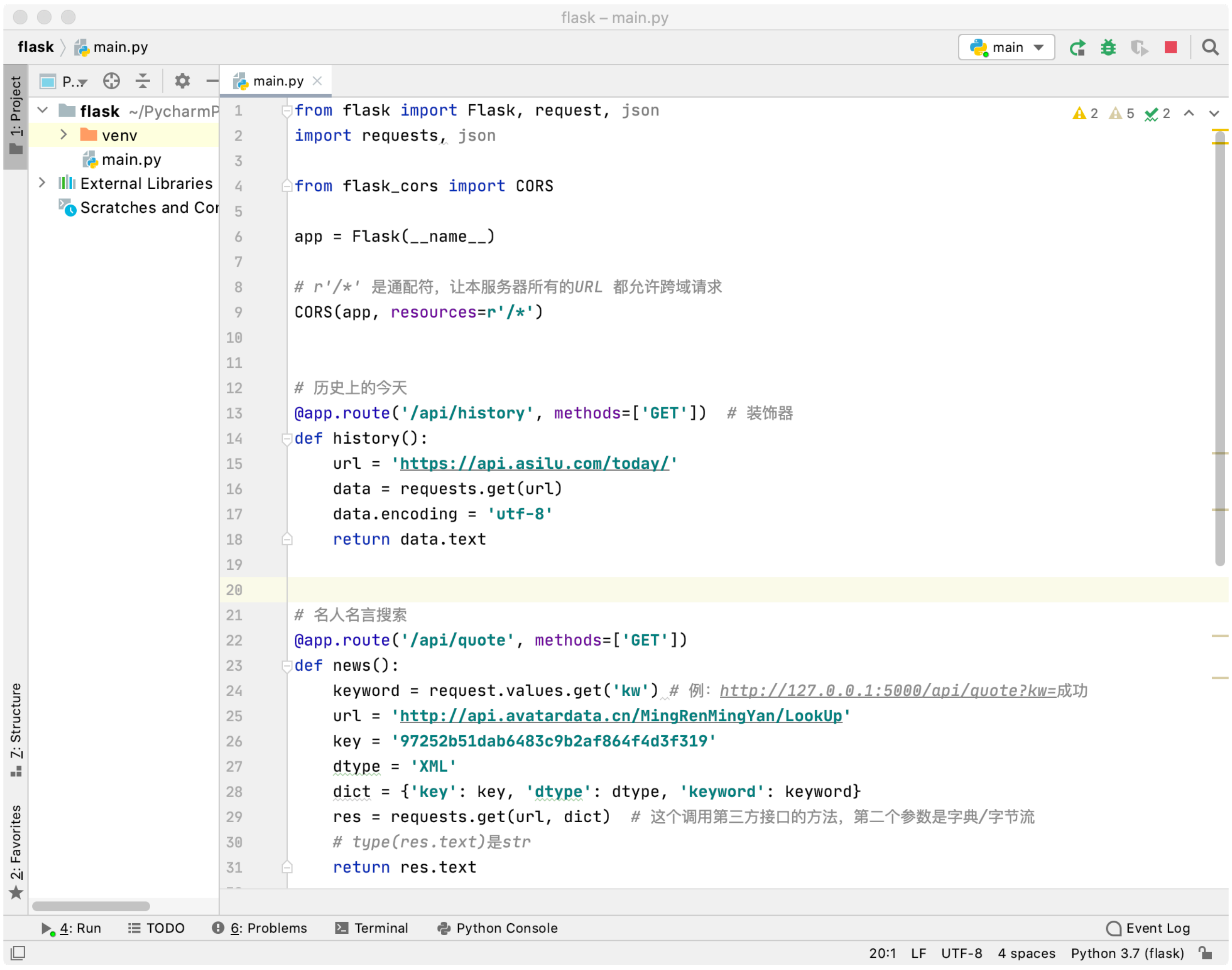
Above allows all URLs on this server to allow cross-domain requests.

Then I bind functions to a route and define the methods of data requsting. If there need to be some parameters I use request.value.get(), then pass on to call for web-apis and return data after encoding (to controll the format). And if there is key needed for a certain web-api (the key should be acquired by myself from their platform), I will wrap the api fine and cut the number of parameters to keep it easy.

```
if __name__ == '__main__':
    app.run(debug=True)
```

Finally app.run. Keep the back-end server working and then test all my api at the front-end. They all work out.

3.2.3 Essential Code Pics



```
from flask import Flask, request, json
import requests, json

from flask_cors import CORS

app = Flask(__name__)

# r'/*' 是通配符，让本服务器所有的URL 都允许跨域请求
CORS(app, resources=r'/*')


# 历史上的今天
@app.route('/api/history', methods=['GET'])  # 装饰器
def history():
    url = 'https://api.asilu.com/today/'
    data = requests.get(url)
    data.encoding = 'utf-8'
    return data.text


# 名人名言搜索
@app.route('/api/quote', methods=['GET'])
def news():
    keyword = request.values.get('kw') # 例: http://127.0.0.1:5000/api/quote?kw=成功
    url = 'http://api.avatardata.cn/MingRenMingYan/LookUp'
    key = '97252b51dab6483c9b2af864f4d3f319'
    dtype = 'XML'
    dict = {'key': key, 'dtype': dtype, 'keyword': keyword}
    res = requests.get(url, dict)  # 这个调用第三方接口的方法，第二个参数是字典/字节流
    # type(res.text)是str
    return res.text
```

```python
# 周公解梦
@app.route('/api/dream', methods=['GET'])
def dream():
    keyword = request.values.get('kw') # 例: http://127.0.0.1:5000/api/dream?kw=黄金
    url = 'http://v.juhe.cn/dream/query'
    key = 'c5179ebd54f625613902bc6660d4b1b5'
    dict = {'key': key, 'q': keyword}
    res = requests.post(url, dict)
    res.encoding = 'utf-8'
    # type(res.text)是str
    return res.text


if __name__ == '__main__':
    app.run(debug=True)
```

4. Apis and Intergration

4.1 All of the web-apis I used

src	BaseURL	Data Format	Note
简爱API	https://api.asilu.com/today	json	历史上的今天
阿凡达数据	http://api.avatardata.cn/MingRenMingYan/LookUp	xml	名人名言查询
Kate·API	https://api.66mz8.com/api/translation.php	json	中英互译
聚合数据	http://ip.taobao.com/service/	json	周公解梦查询

4.2 My Apis

Method	Api	Note
GET	/api/history	Get today's history affairs.
GET	/api/quote?kw	Get the relevant famous sayings list searched by a certain keyword.
GET	/api/dream?kw	Get the detaied dream analysis list searched by a certain keyword.

4.3 Intergration

Back-end:

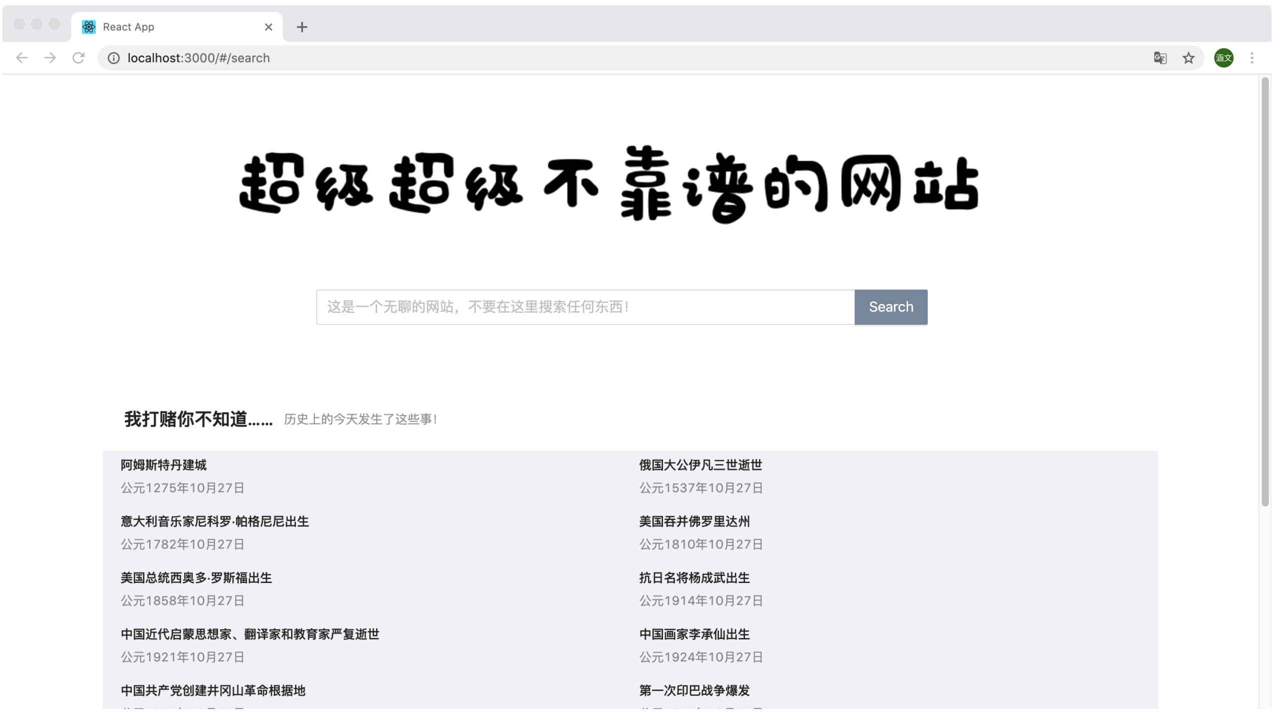
```
@app.route('/api/dream', methods=['GET'])
def dream():
    keyword = request.values.get('kw') # 例: http://127.0.0.1:5000/api/dream?kw=
黄金
    url = 'http://v.juhe.cn/dream/query'
    key = 'c5179ebd54f625613902bc6660d4b1b5'
    dict = {'key': key, 'q': keyword}
    res = requests.post(url, dict)
    res.encoding = 'utf-8'
    # type(res.text)是str
    return res.text
```

Front-end:

```
      var url = '/dream?kw=' + this.state.keyword
        axios.get(url).then((res) => {
          console.log('res3:', res.data)
          var result=res.data.result
          this.setState({myData:result})
      })
```

```
    render() {
      //初始化render数组状态
      let objArr=this.state.myData
      if(objArr!==null){
        return(
          <Layout>
              <PageHeader
                  className="site-page-header"
                  ghost={false}
                  title="最后我还熟读《周公解梦》, 昨晚梦见这个啦? "
                  subTitle="想不到吧"
              />
            <List
                itemLayout="vertical"
                dataSource={objArr}
                style={{ margin: '0 20px 0 20px' }}
                split={true}
                renderItem={item => (
                <List.Item>
                    <List.Item.Meta
                    title={'如果你梦到"'+item.title+'"'}
                    description={item.des}
                    />
                    <Divider />
                </List.Item>
                )}
            />
          </Layout>
        )
      }
      else return(
        <Layout>
        <PageHeader
          className="site-page-header"
          ghost={false}
          title="最后我还熟读《周公解梦》，昨晚梦见这个啦？"
          subTitle="想不到吧"
      />
        <Empty
            style={{margin: '10px'}}
            description="被你逮到了，我确实不知道"
        >
        </Empty>
        </Layout>
      )
    }
```

5. Screenshots



我打赌你不知道…… 历史上的今天发生了这些事！

阿姆斯特丹建城
公元1275年10月27日

俄国大公伊凡三世逝世
公元1537年10月27日

意大利音乐家尼科罗·帕格尼尼出生
公元1782年10月27日

美国吞并佛罗里达州
公元1810年10月27日

美国总统西奥多·罗斯福出生
公元1858年10月27日

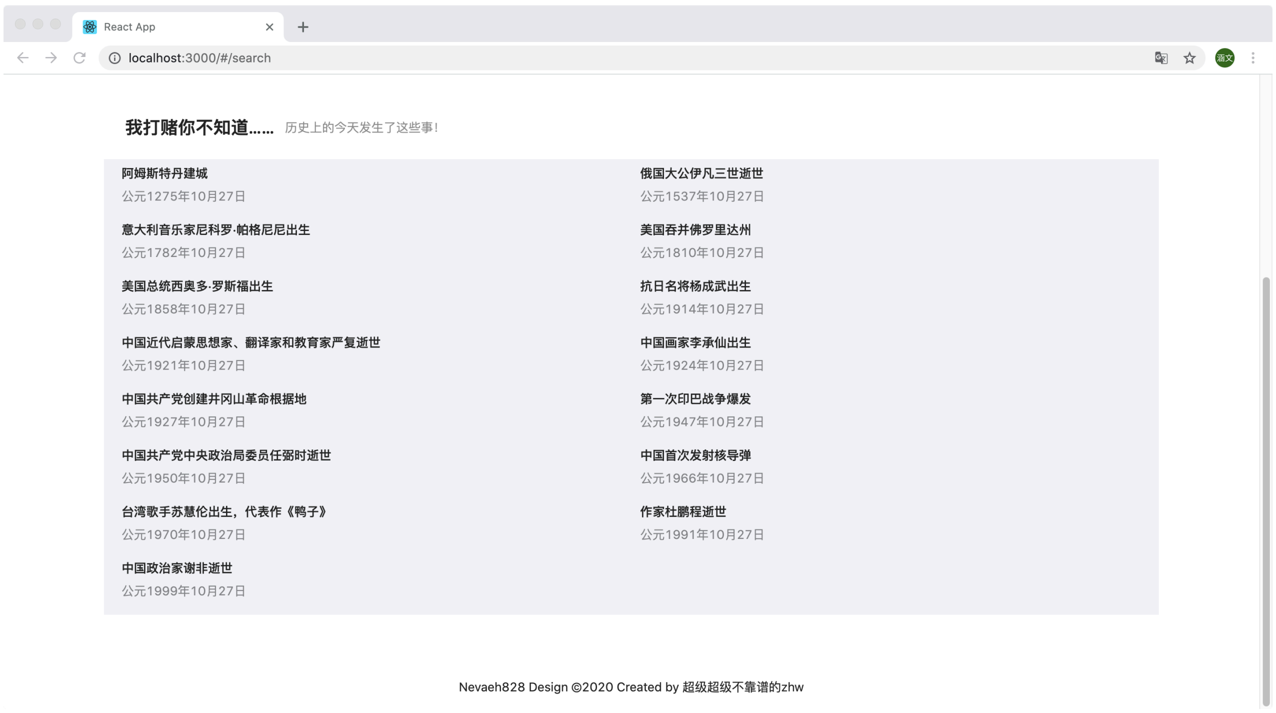
抗日名将杨成武出生
公元1914年10月27日

中国近代启蒙思想家、翻译家和教育家严复逝世
公元1921年10月27日

中国画家李承仙出生
公元1924年10月27日

中国共产党创建井冈山革命根据地

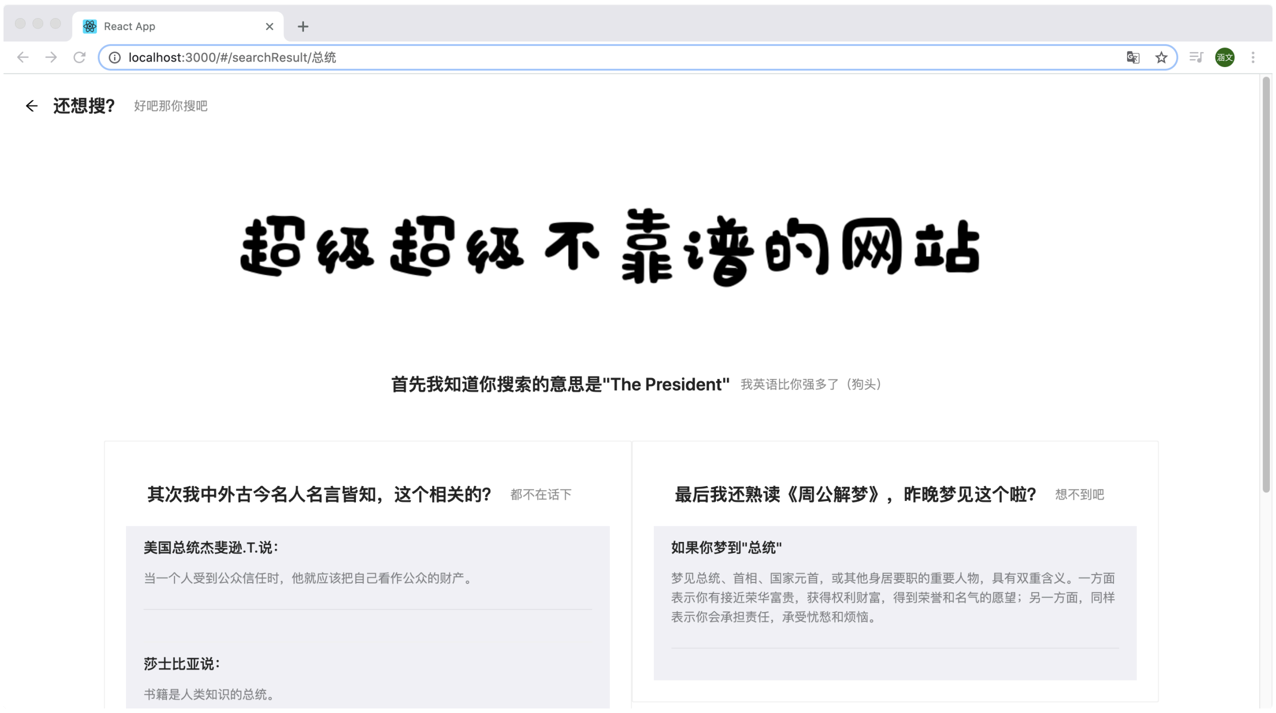
第一次印巴战争爆发



我打赌你不知道…… 历史上的今天发生了这些事!

阿姆斯特丹建城 公元1275年10月27日	俄国大公伊凡三世逝世 公元1537年10月27日
意大利音乐家尼科罗·帕格尼尼出生 公元1782年10月27日	美国吞并佛罗里达州 公元1810年10月27日
美国总统西奥多·罗斯福出生 公元1858年10月27日	抗日名将杨成武出生 公元1914年10月27日
中国近代启蒙思想家、翻译家和教育家严复逝世 公元1921年10月27日	中国画家李承仙出生 公元1924年10月27日
中国共产党创建井冈山革命根据地 公元1927年10月27日	第一次印巴战争爆发 公元1947年10月27日
中国共产党中央政治局委员任弼时逝世 公元1950年10月27日	中国首次发射核导弹 公元1966年10月27日
台湾歌手苏慧伦出生，代表作《鸭子》 公元1970年10月27日	作家杜鹏程逝世 公元1991年10月27日
中国政治家谢非逝世 公元1999年10月27日	

Nevaeh828 Design ©2020 Created by 超级超级不靠谱的zhw



还想搜? 好吧那你搜吧

超级超级不靠谱的网站

首先我知道你搜索的意思是"The President" 我英语比你强多了（狗头）

其次我中外古今名人名言皆知，这个相关的? 都不在话下

美国总统杰斐逊.T.说:

当一个人受到公众信任时，他就应该把自己看作公众的财产。

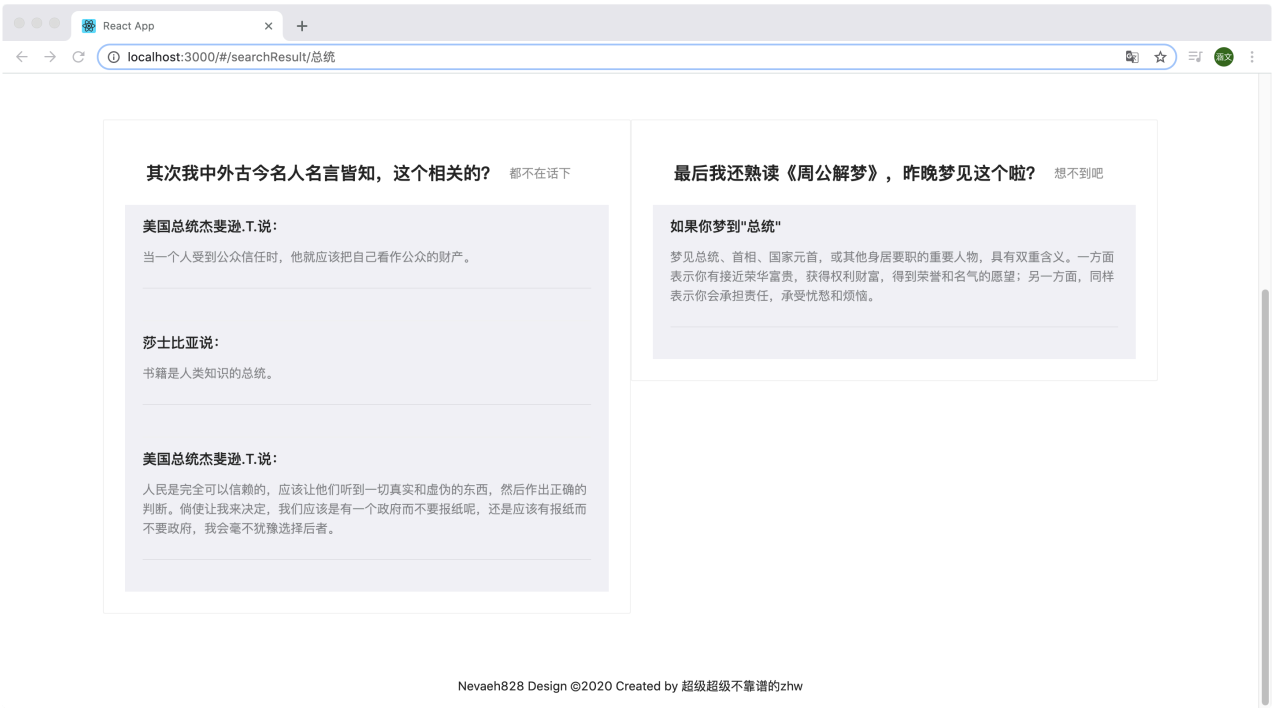
莎士比亚说:

书籍是人类知识的总统。

最后我还熟读《周公解梦》，昨晚梦见这个啦? 想不到吧

如果你梦到"总统"

梦见总统、首相、国家元首，或其他身居要职的重要人物，具有双重含义。一方面表示你有接近荣华富贵，获得权利财富，得到荣誉和名气的愿望；另一方面，同样表示你会承担责任，承受忧愁和烦恼。



其次我中外古今名人名言皆知，这个相关的? 都不在话下

美国总统杰斐逊.T.说：

当一个人受到公众信任时，他就应该把自己看作公众的财产。

莎士比亚说：

书籍是人类知识的总统。

美国总统杰斐逊.T.说：

人民是完全可以信赖的，应该让他们听到一切真实和虚伪的东西，然后作出正确的判断。倘使让我来决定，我们应该是有一个政府而不要报纸呢，还是应该有报纸而不要政府，我会毫不犹豫选择后者。

最后我还熟读《周公解梦》，昨晚梦见这个啦? 想不到吧

如果你梦到"总统"

梦见总统、首相、国家元首，或其他身居要职的重要人物，具有双重含义。一方面表示你有接近荣华富贵，获得权利财富，得到荣誉和名气的愿望；另一方面，同样表示你会承担责任，承受忧愁和烦恼。

Nevaeh828 Design ©2020 Created by 超级超级不靠谱的zhw

6. Background Setup Instruction

Front-end background

```
npm -v
yarn -v
node -v
git --version

# antd
npm i antd
yarn add craco-less

# 路由
npm i react-router-dom

# Proxy 代理
yarn add http-proxy-middleware

# 启动
cd Web_Service_SOA
cd CrazyPages
npm start
```

Back-end background

```
pip install flask
pip install requests
pip install flask_cors
......

python flask.py
```

By the way, the project has been tracked in my GitHub repository(link below). You can see my follow-up development and improvement.

https://github.com/Nevaeh828/Web_Service_SOA