

# Report for Final Project

(Group 6)

## Group Information

Student 1	
Student Name	Pan Wenzheng (潘文峥)
Student Number	520030910232
Class	F2003002
Contribution	<ul style="list-style-type: none"> <li>✓ Contriving the general plan</li> <li>✓ Crawling all pet data</li> <li>✓ Building the <i>All-kinds</i> and <i>Detail</i> pages</li> </ul>
Student 2	
Student Name	Lai Yuejia (赖粤嘉)
Student Number	520030910230
Class	F2003002
Contribution	<ul style="list-style-type: none"> <li>✓ Contriving the general plan</li> <li>✓ Crawling all service data</li> <li>✓ Building the <i>Service</i> pages</li> </ul>
Student 3	
Student Name	Zhu Yanhong (朱炎虹)
Student Number	520030910203
Class	F2003002
Contribution	<ul style="list-style-type: none"> <li>✓ Contriving the general plan</li> <li>✓ Building the <i>Home</i> and <i>About</i> pages</li> <li>✓ Designing illustrations</li> </ul>
Student 4	
Student Name	Liu Sijia (刘思嘉)
Student Number	520030910200
Class	F2003002
Contribution	<ul style="list-style-type: none"> <li>✓ Contriving the general plan</li> <li>✓ Proceeding data visualization and building the <i>Visualization</i> page</li> </ul>
Student 5	
Student Name	Wang Huanbo (王焕博)
Student Number	520030910206
Class	F2003002
Contribution	<ul style="list-style-type: none"> <li>✓ Contriving the general plan</li> <li>✓ Building the <i>All-kinds</i> page</li> </ul>

## Abstract

In decent accord with the requirement assigned, the group have endeavored to reach constructive consensus, opting for the theme of final project, building a website for pets. Thereafter, a compact website is to be erected, one that have combined the application of techniques acquired in the course during the semester, i.e., data obtainment (through internet crawlers), database implementation, text retrieval and data visualization.

Having contemplated above, the group worked jointly on the construction of the website that is focused on illustration of pet information with exclusive clarity and credibility, whereby intelligible data turn accessible to pet-lovers as functionality.

It should be noted that this report is to state whatever warrant reference within the completion of the project.

## Key Words

Website construction, data crawler, text retrieval, data visualization, database application

# Contents

<b>I. Project Design</b>	<b>4</b>
<b>II. Project Function</b>	<b>6</b>
2.1 <i>Home Page</i>	6
2.2 <i>All-kinds Pages</i>	9
2.3 <i>Service Pages</i>	11
2.4 <i>Data Visualization Pages</i>	12
2.5 <i>About Pages</i>	13
<b>III. Function Realization</b>	<b>15</b>
3.1 <b>Data Obtainment</b>	15
3.1.1 Section 1: <i>All-kinds</i> Part	15
3.1.2 Section 2: <i>Service</i> Part	22
3.2 <b>Website Construction</b>	23
3.2.1 <i>Home Page</i>	23
3.2.2 <i>All-kinds Pages</i>	28
3.2.3 <i>Service Pages</i>	31
3.2.4 <i>About Pages</i>	35
3.3 <b>Data Visualization</b>	36
3.3.1 Pet Map	36
3.3.2 Feature Statistics	41
3.4 <b>Text Retrieval</b>	43
<b>IV. Concluding Remarks</b>	<b>44</b>
<b>V. Acknowledgement</b>	<b>44</b>

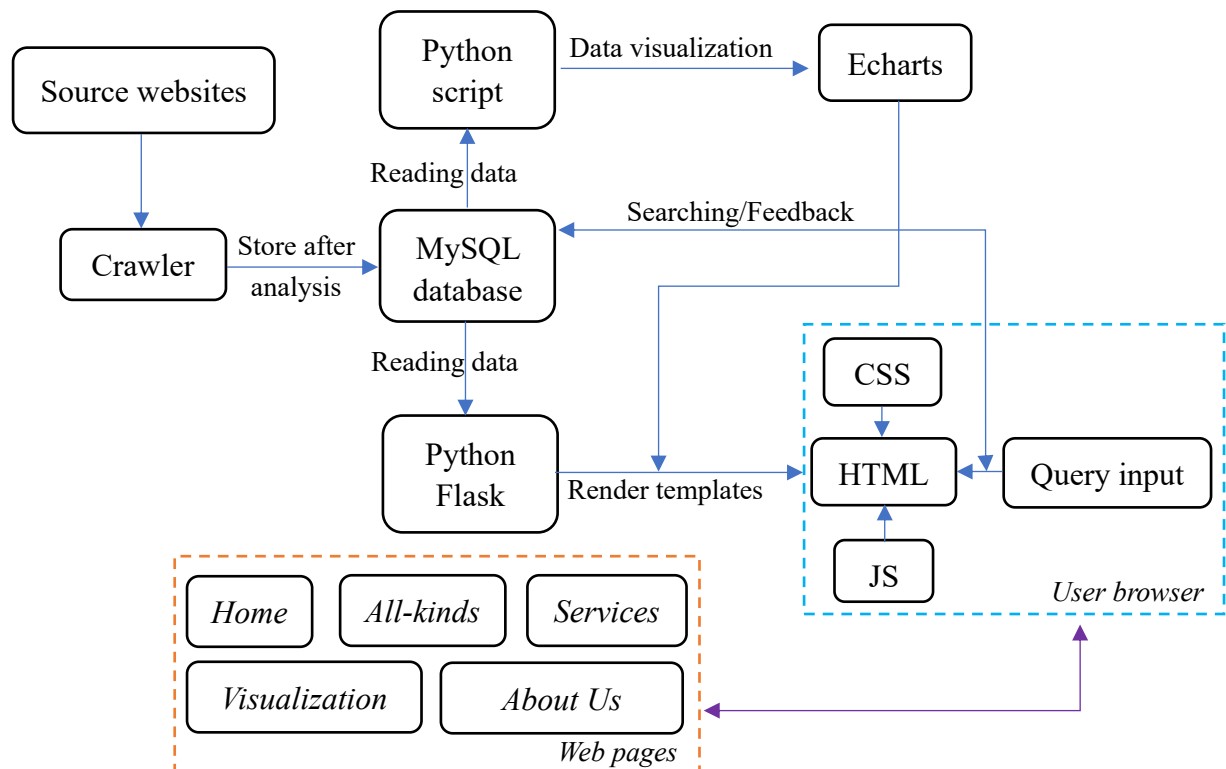
## I. Project Design

As final project of this course, the principles we have focused on about design have pivotal significance to the subsequent work. In narrow sense, the website to be built is supposed to be a demonstration of application of techniques we have learned during the course, and equally importantly, we aimed to work out this project in terms of realistic demands we investigated when looking for appropriate websites as source in much broader sense.

To pinpoint the former, we have elaborately divided labor in accord with the respective proficiency of group members, which results in a reasonable as well as efficient cooperative working process. To be specific, two of us take charge in the data obtainment using crawlers, separately build the *All-kinds* and *Services* pages and co-implemented the retrieval part, one of us shoulders the responsibility to design illustrations and co-build the *home* page, one of us separately completed data visualization and build the *visualization* page, and all of us contributed to the formation of this overall plan for design, which this part mainly includes.

Concerning the latter perspective, the website we are building is committed to providing accurate and ordered information for pet-lovers to an extend of clarity that sites in being hardly approach. By removing annoying advertisements and meaningless articles without credibility, our website is aimed at re-organizing and re-structuring information we gained in forms of cards, lists and graphs etc., immensely beautifying the interface, while realizing comprehensive augmentation as functionality.

The diagram below depicts how this project is arranged and proceeded:



## 1.1 Original Designing Intention

In order to help many pet breeders and pet lovers understand more pet types, pet breeding knowledge, enhance the awareness of protecting pets and enhance the sense of pet keeping belief, our group has built a pet website with temperature, contents and prospects.

## 1.2 Project Highlights

### (1) More beautiful web pages

When crawling the website data information, we removed the information corresponding to the advertising and marketing articles, improved the quality of the website content and improved the user experience.

### (2) More reasonable web page structure

We use a more reasonable navigation bar layout, and the information division of each functional web page is clearer, which is convenient for users to query information.

### (3) Richer content

We have integrated the information of multiple websites, increased the total number of pets and expanded other query functions. Such as the enrichment of pet service.

## 1.3 Project Function Division

The main functions of this website are divided into Home Page, All-kinds Pages, Services Pages, Data Visualization Page and About Page.

The functions of each module are relatively independent and complete; At the same time, there is a certain inline relationship between each module, which enhances the integrity of the project.

## 1.4 Space for Promotion

### (1) Search function

The implementation of search function can be more comprehensive. For example, the navigation bar on the home page can expand the search function to the whole website, rather than just for specific pet types in All-kinds Page.

### (2) Interactive function

The website can also expand some interactive functions. In view of the characteristics of pet breeding, some channels for users to communicate can be created on the basis of the current website, so as to improve the richness and practicability of the project.

## II. Project Function

### 2.1 Home Page

In Home Page, we want to show show the all functions of this web and offer a search table to search.

#### (1) Navigation bar

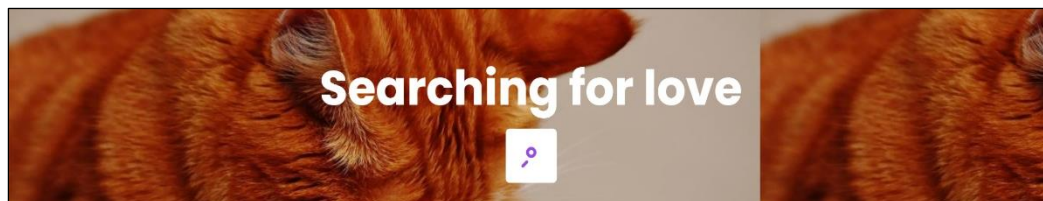


In the navigation bar, we divide the functions of the whole website into five parts. As shown in the figure above. Web users can find and obtain specific information by clicking the corresponding part of the navigation bar, which is convenient for users and beautifies the layout of the website.

In addition, we also set a sub menu, as shown in the figure below. It is more convenient for users to obtain dedicated information, and makes the web page layout more reasonable.



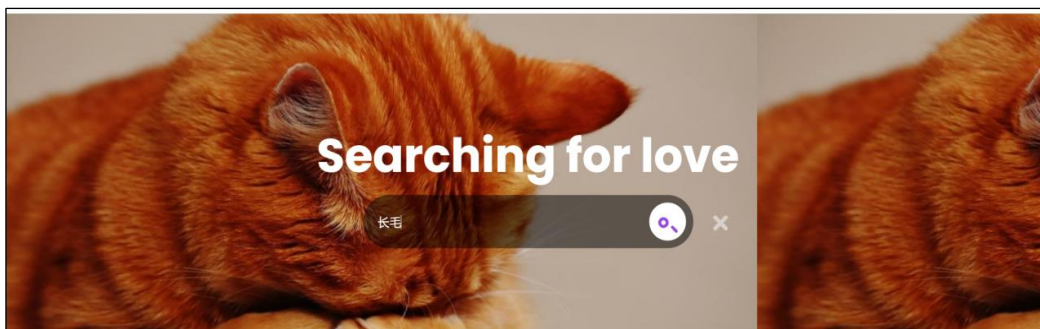
#### (2) Searching function



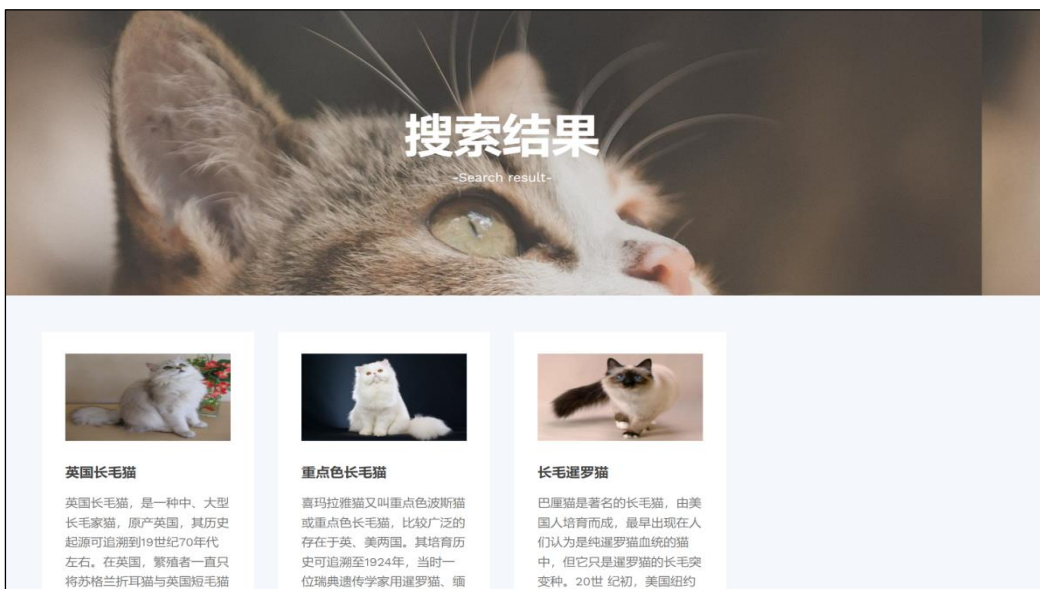
The search function of the home page is mainly aimed at the contents of All kinds Page. For example, you can search specific pet types or some pet keywords, as shown in the figure below. Through this search box, users can obtain the specific information of

corresponding pets without deliberately jumping to the All kinds Page for search, which is convenient for users.

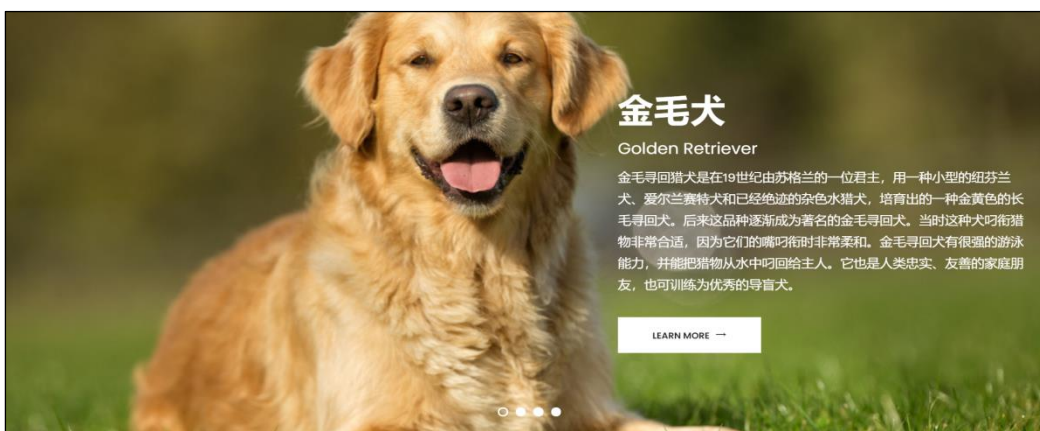
For example, we choose the key word “长毛” to search.



The below result will be returned:



### (3) Typical pet display



In the third part of the home page, we choose to show some typical pets. They are:“金



毛犬”、“哈士奇”、“波斯猫”、“加菲猫” In this section, users can learn about a typical pet, and click the LEARN MORE button to enter the ALL KINDS page to learn its details. The purpose of setting this part is to improve users' browsing interest, enrich web content, and strengthen the links within the website.

#### (4) Information expansion



In order to enrich the content of the website and sublimate the significance of the establishment of the website, we added the information expansion part at the end of the home page to list the information about the current situation of pet rearing (The three parts are “养宠现状”、“养宠益处”、“宠物趣图”). Users can click the link on LEARN MORE button to jump to the Internet to understand the rich information.

The display is as follows: if you want to learn more about “养宠现状”, then you can click on the button which will turn to page below, which you can learn more details.

**2021年中国宠物行业市场前景及投资研究报告**

中商产业研究院  
已认证账号

80 人赞同了该文章

宠物一般是指家庭饲养的、作为伴侣动物的狗、猫、淡水观赏鱼、鸟、爬行动物等。国内宠物行业起步相对较晚，但随着人们经济水平的提高、思想观念的转变以及人口老龄化和单身、丁克群体的增长，饲养宠物成为人们的一种情感寄托，人们从宠物身上获取温暖和陪伴，宠物经济由此快速发展。

**一、宠物行业分类**

宠物产业是指涉及与宠物相关的所有领域，主要包括宠物食品、宠物医疗、宠物培训、宠物饲养、宠物用品、宠物美容、宠物保险等，具体细分情况如下：

宠物消费行业		细分市场
产品类	宠物饲养	宠物养殖、宠物交易等
	宠物食品	宠物干粮、宠物湿粮、宠物零食、宠物营养品等
	宠物用品	宠物玩具、宠物服装、清洁工具等
	宠物医疗	病情诊疗、疾病防御、宠物保健等
服务类	宠物美容	洗澡、剪毛、剪指甲、染毛等

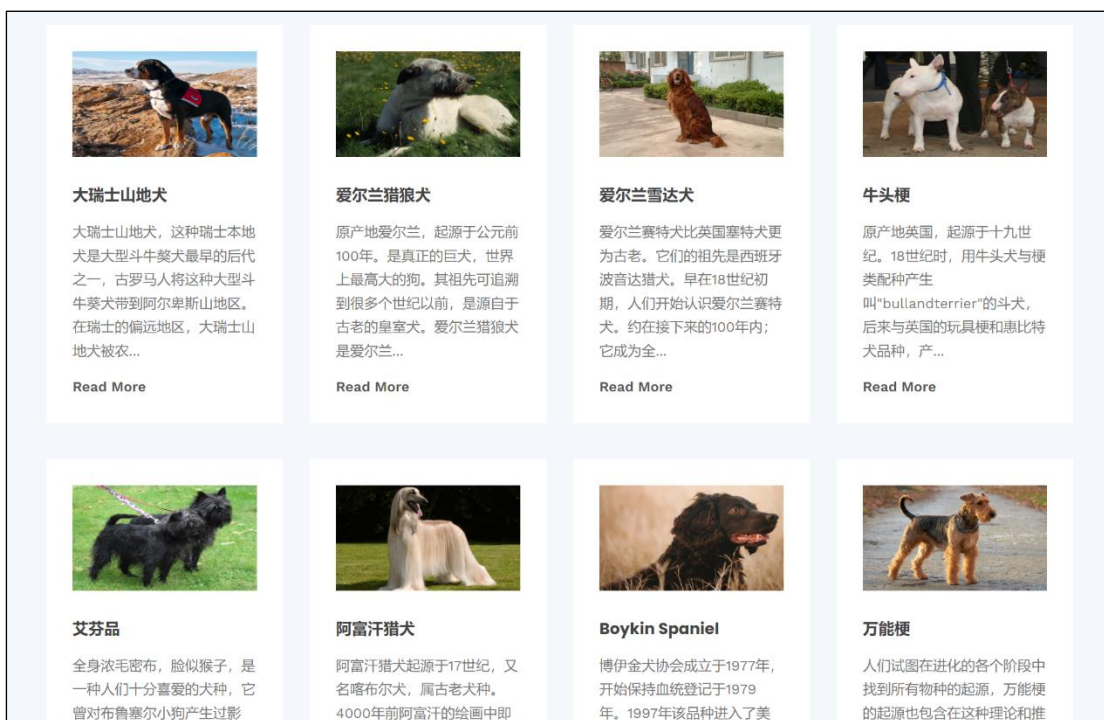


## 2.2 All-kinds Pages

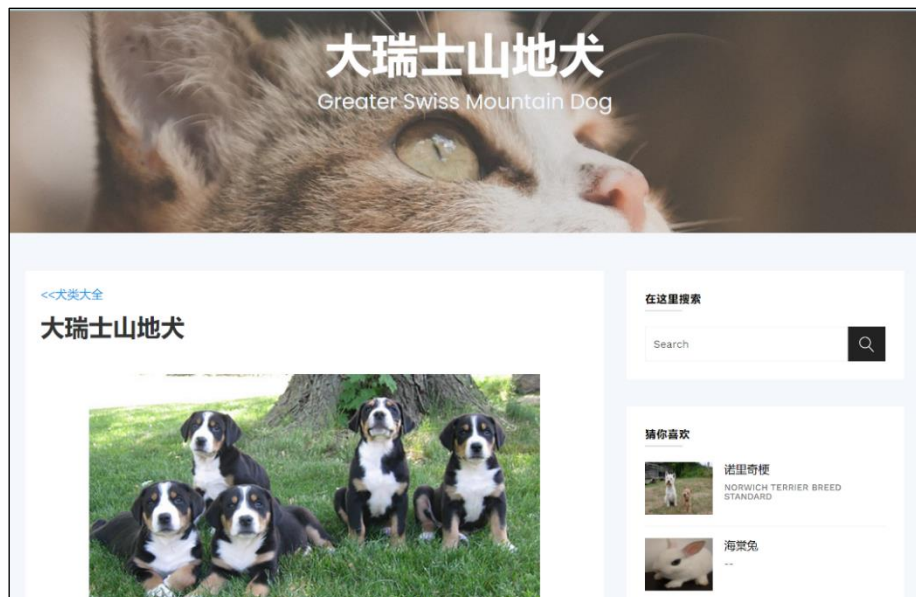
This page is aimed at displaying all information of animals we have get and searching for the animals you want.



Like the classic dictionary look-up mode, the *All-kinds* page gives labels for various animals, such as cats, dogs, etc. in the various labels, it is further refined to each animal, so as to simplify the search. For each animal, we have attached pictures and short text descriptions to the page, and corresponding links.



In the link, we will display the details of the corresponding pets obtained from the database, and the quantifiable data such as life span and weight will be displayed through visualization with radar charts. In the corresponding links, we will also give other pets with the same characteristics as this type of pet, so that you can choose according to your preferences.



**注意事项**

大瑞士山地犬相关疾病 疾病的征兆 生病的狗会变得比较迟钝, 不活跃, 食欲减少或增加。 紧急的警讯: 狗若有下列征兆请立即送到兽医院: 1. 衰竭。 2. 连续呕吐超过 24 小时。 3. 下痢超过 24 小时。 4. 呼吸困难。 5. 口、鼻、耳、肛门、尿道流血。 6. 显著疼痛。 生病的主要征兆 1. 苍白无血色。 2. 呕吐。 3. 下痢。 4. 呼吸困难。 5. 流血。 6. 搔抓不安。 7. 精神萎靡。 8. 无食欲。 其它生病的一般征兆 1. 咳嗽。 2. 耳痛。 3. 耳翼肿胀。 4. 耳朵有分泌物。 5. 眼睛肿胀, 有分泌物。 6. 跛行。 7. 皮肤瘙痒、红肿、脱毛。 8. 粪尿失禁。 9. 便秘。 测量体温 正常狗的体温介于摄氏 38 度到 39 度, 兴奋时体温约为上升, 体温超过 39℃ 以上都是不正常的, 必须告知兽医。 测体温时可用粗扁型动物用温度计, 比较不易折断。 先将温度计, 甩一甩使水银降到 35℃ 以下。 找一个人将狗捉住, 温度计涂上润滑剂后, 一手将尾巴举起, 一手将温度计插入肛门, 朝上, 使温度计紧靠直肠壁。 大约 30 秒, 取出温度计, 读温度计并记录之。 温度计清洗消毒后收存备用。 新型人用数字式温度计也可用于犬只体温测量。 测量脉搏 脉搏代表心跳, 最易测量脉搏位置在大腿内侧的股动脉, 将手指压住股动脉, 感觉到脉搏后开始计数计时。 小型狗的心跳每分钟 90~120 次大型狗每分钟 65~90 次。 测量呼吸数 狗在休息状态下, 正常呼吸数每分钟 10~30 次 (小狗比大狗呼吸数高), 健康正常的狗呼吸平顺。 当你的狗出现上述一项或多项疾病征兆时, 请按下列疾病征兆追查检查记录之, 常见疾病征兆分为: 1. 苍白无血色。 2. 呕吐。 3. 下痢。 4. 呼吸不正常。 5. 出血。 6. 皮肤瘙痒。 依上述 6 项以追踪问答方式请畜主检查记录之, 将狗送到兽医诊所时, 一并交给兽医, 协助兽医正确快速的了解你的狗的问题。

**我的留言**

Your Comments

发表评论

If you don't have enough time to browse one by one, we also provide a search bar that can directly query the name of your pet to help you get the information you want as soon as possible.

## 2.3 Service Pages



Pet service web page will be a series of articles about pets classified and displayed, which are mainly divided into four aspects of pet medical treatment, pet training, pet funny and pet maintenance, each part has a corresponding search.



The details of each article page and related article recommendations, can allow users to find the information they want more accurately and effectively, for each page, we placed nine articles, and at the bottom of each page to achieve a page turning function, so that the layout of the entire website is more reasonable. Of course, like other web pages, we still set up a navigation bar above the web page to facilitate users to jump between web pages, and for the convenience of users, we have set up jump links on pictures and corresponding parts of the text.

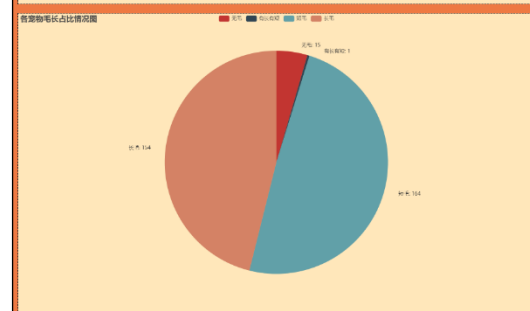
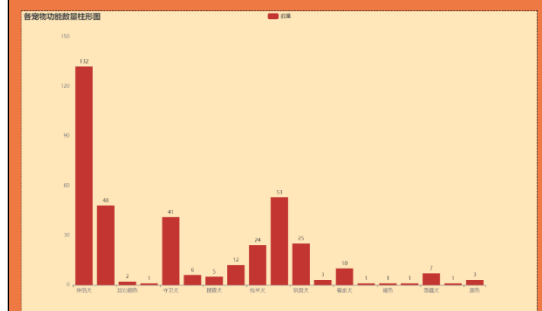
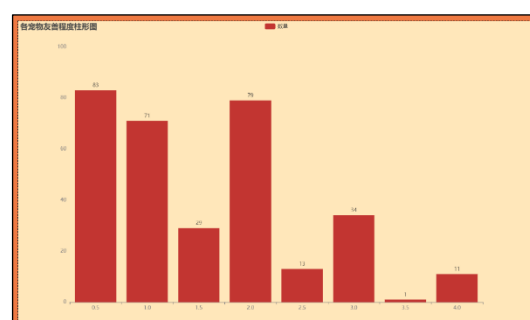
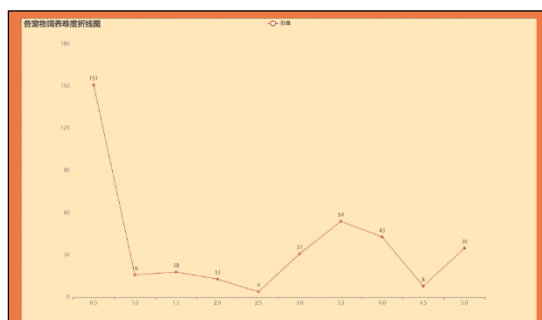
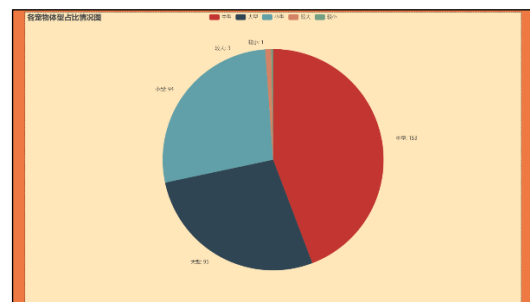
And in the details page of each article, we have also set up searches and corresponding return links and other links.



## 2.4 Data Visualization Pages

This section consists of two sub-pages, and they are a world map for pets as well as charts showing pets' characteristics.

Below shows the demonstration of these pages.





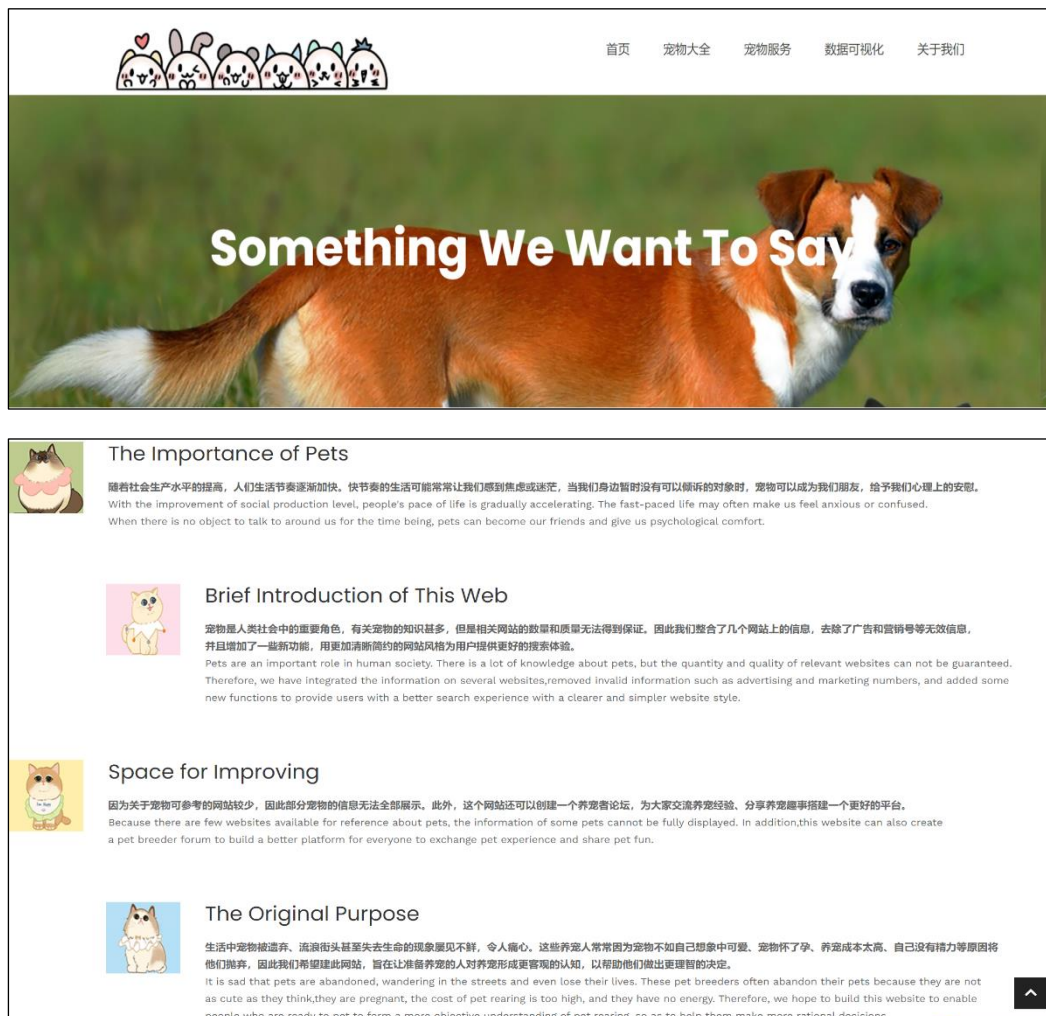
## 2.5 About Pages

In *About* Page, we set two parts. In the first part we say something to reflect the reality and make an appeal, in the second part we display the job contributions between team members.

### (1) Something we want to say

In this section, we give some discussions on pet breeding, and also point out some room for progress in this website. The purpose is to enhance pet breeders' sense of feeding belief and give users some space for thinking in addition to the objective functions of the website.

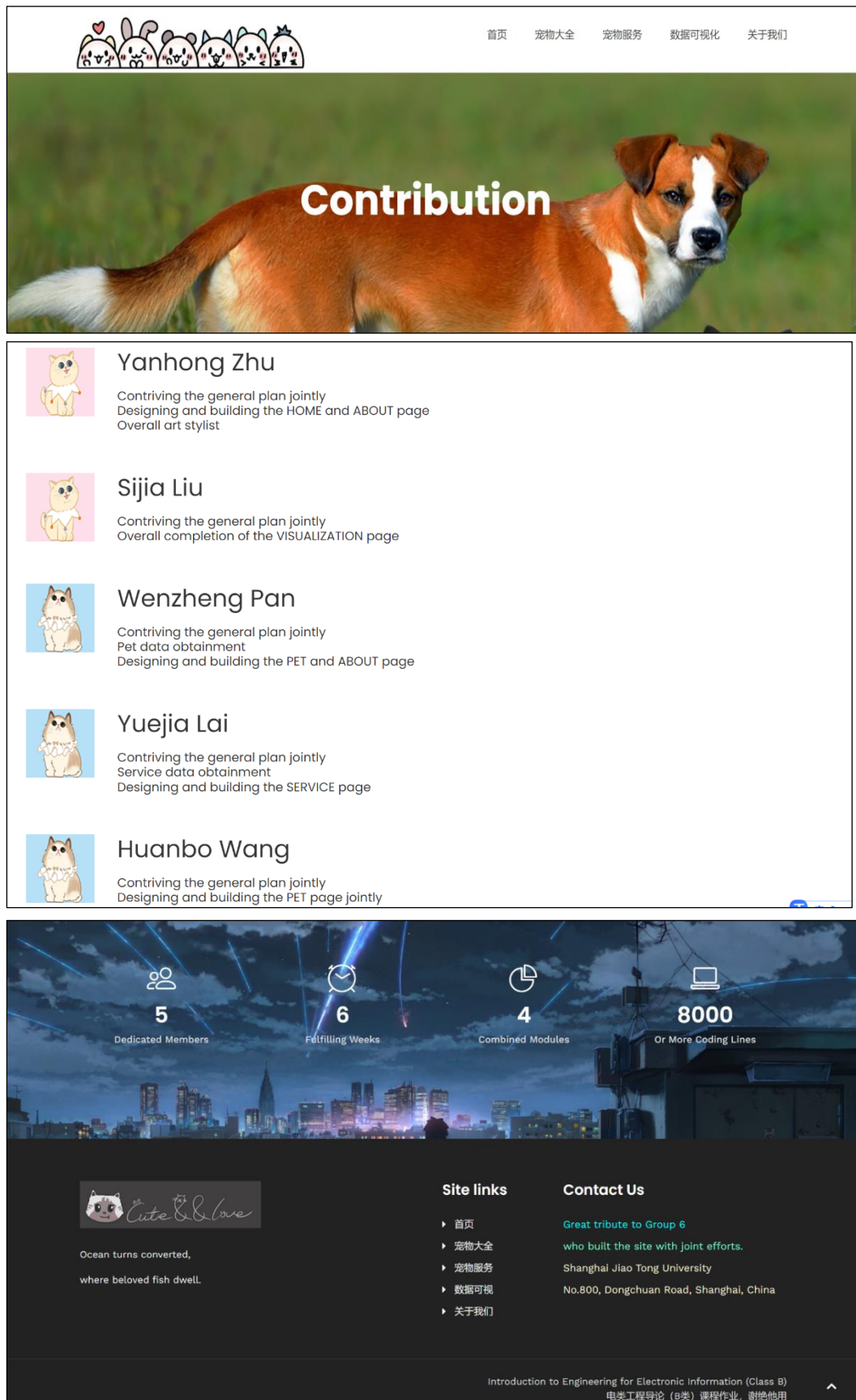
The display is as follows:



### (2) Contribution

In this part, we show the division of labor of the project. It can be seen that every member of the team has played an important role in the implementation of the project,

which is the main reason for the completion of this website.





## III. Function Realization

### 3.1 Data Obtainment

This part reviews the key points, difficulties and core technique concerning the acquisition of data required, namely web crawler and database application, whereby the entire data get captured to erect the website.

Basically, all the data our website is to consist of have been extracted from two sites: <http://www.chongqu.com.cn> and <http://www.mengchong.cn>, and the desired data range from detailed introduction of each pet to the services for raising pets.

It should be noted that the programming tool of this part of task is generally comprised of external python libraries: *requests* (to generate requests to URLs), *pymysql* (to command database) and *lxml* (where *etree* is imported to decode html texts and retrieve tags).

Equally worth mentioning is the data scale of this project. Moderate as the number of pet items may seem, focus is supposed to be concentrated on the considerable field names each item contains and that the complexity to have the 34 names of a single pet out of over a thousand should not be neglected, and articles concerning pet services which have been categorized into 5 columns also reaches a pronounced magnitude of thousands of data items. Additionally, extra remarkable effort to extract the longitude as well as latitude of the origins of each pet to generate the *pet-map* should also count.

In light of the above remarks, it is obligated to state that **20,000 to 30,000** is relatively a rough but appropriate evaluation of the overall magnitude of *data scale* when assessing the *workload* of this project.

#### 3.1.1 Section 1: *All-kinds Part*

Prior to proceeding the crawlers, it is of utmost significance to form an explicit sequence of subtasks and have them done in turn. Taking this mentality into account, the following heads of functions are preliminarily formulated, which manifests the basic procedures through which this part of project is to be implemented:

```
1. #从网站 1 获得每个宠物具体信息所在的 url
2. def getUrls(page,headers=headers):
3. #...
4.
5. #从网站 2 获得每个宠物具体信息所在的 url
6. def getUrls_2():
7. #...
8.
9. #从网站 1 获取猫和犬类信息
```

```
10. def getData_cat_and_dog(url,headers=headers):
11. #...
12.
13. #从网站 2 获取其他宠物信息
14. def getData_others(url):
15. #...
16.
17. #将信息写入数据库操作
18. def insertData(datalist):
19. #...
20.
21. #封装执行函数 1
22. def act_data1_to_db():
23. #...
24.
25. #封装执行函数 2
26. def act_data2_to_db():
27. #...
28.
29. #调用执行函数，完成从爬取信息到写入数据库全过程
30. act_data1_to_db():
31. #...
32. act_data2_to_db():
33. #...
```

Noting that through preliminary analysis of the web structure, it is required to find out the position the targeted data is organized and store them in an orderly format. Evidently, the core part of this task lies in the process of analyzing the structure the web html is organized and locate the desired data through appropriate methods.

To ensure successful batch operation of access to our target pages, user-agents and referrer used when requested by real browser (e.g., Microsoft Edge) are duplicated as a dictionary named headers and to be passed as a parameter to the “get” functions mentioned above.

```
1. headers = {
2.     "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) \
3.         AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54 \
4.         Safari/537.36 Edg/95.0.1020.30",
5.     "Referer": "http://www.chongqu.com.cn/chongwudaquan/"
6. }
```

Specific analysis and realization of the procedures mentioned above is to be displayed as follows. Note that due to the similarity between the two source sites chosen and in case of lengthiness and repetition, operation over the former alone shall exemplify the entire proceeding for the time being.

Noting that the primary page which lists all the cards for pets is organized in several pages and analyzing the constitution of the URL for each page, it is noticeable that page information has been contained in the suffix indicating the relative path leading to them (e.g., [http://www.chongqu.com.cn/chongwudaquan/list\\_22\\_4.html](http://www.chongqu.com.cn/chongwudaquan/list_22_4.html) is the forth page of all dogs).

Subsequently, focus is concentrated on the extraction of URLs for the detailed information of each specific pet species. By inspection, the place where such links dwell is obtained, in a `<p>` tag whose class name is "info-t" in particular. Hence, the *xpath* expression for locating the link can be derived: `//p[@class="info-t"]/a/@href'`.

```
<p class="info-t">
  <a href="/chongwudaquan/556.html" target="_blank" class="j-link">土耳其梵猫</a> == $0
</p>
```

So far, the first function by which URLs for detailed information is completed:

```
1. #从网站 1 获得每个宠物具体信息所在的 url
2. def getUrls(page,headers=headers):
3.     url = f"http://www.chongqu.com.cn/chongwudaquan/list_22_{page}.html"
4.     response = requests.get(url, headers=headers)
5.     response.encoding = response.apparent_encoding
6.     tree = etree.HTML(response.text)
7.     return tree.xpath('//p[@class="info-t"]/a/@href')
```

Then we have reached the most significant as well as the most challenging phase of data obtainment, namely to extract the detailed information of each pet and appropriately adjust the format of them so that database can store them at its disposal.

金毛犬			
基本资料			
中文名:	金毛犬	别名:	金毛;金毛寻回猎犬;黄金猎犬;黄金拾猎犬
体型:	大型	毛长:	长毛
英文名:	Golden Retriever	智商:	第 4 名
原产地:	英国	体重:	27~34公斤
寿命:	10~15年	肩高:	55~61厘米
毛色:	黄色	功能:	猎枪犬,导盲犬
详细特征			
体味程度:	♥♥♥♥♥	运动量:	♥♥♥♥♥
耐热程度:	♥♥♥♥♥	护家程度:	♥♥♥♥♥
耐寒程度:	♥♥♥♥♥	美容程度:	♥♥♥♥♥
友善程度:	♥♥♥♥♥	口水程度:	♥♥♥♥♥
粘人程度:	♥♥♥♥♥	掉毛程度:	♥♥♥♥♥
喜叫程度:	♥♥♥♥♥	可训程度:	♥♥♥♥♥
饲养难度:	♥♥♥♥♥	关爱需求:	♥♥♥♥♥

基本信息	品种特点	养宠知识	注意事项
<h3>金毛犬发展起源</h3>  <p>金毛寻回猎犬是在19世纪由苏格兰的一位君主，用一种小型的纽芬兰犬、爱尔兰赛特犬和已经绝迹的杂色水猎犬，培育出的一种金黄色的长毛寻回犬。后来这品种逐渐成为著名的金毛寻回犬。当时这种犬叼衔猎物非常合适，因为它们的嘴叼衔时非常柔和。金毛寻回犬有很强的游泳能力，并能把猎物从水中叼回给主人。它也是人类忠实、友善的家庭朋友，也可训练为优秀的导盲犬。</p> <p>金毛寻回猎犬体格健壮，工作热心，可以用来捕捉水鸟，任何气候下都能在水中游泳。深受猎手的喜爱，主要被作为家犬饲养，为大型犬。</p>			

The above pictures demonstrate the original configuration of the detailed pet features, which are placed without explicit structure and exhibited in completely different forms (i.e., short terms are displayed in lists, several characteristics shown in visualized graphs, and long texts divided into paragraphs), thus generating immense difficulty for identifying and locating each of them.

Equally vitally, the application of the except-handling mechanism is imperative for a small number of detailed data are incomplete. Codes bellow show how we tackled different field names of the characteristics of pets<sup>1</sup>, where “--” takes the places where absence of information occurs.

```

1. #从网站 1 获取猫和犬类信息
2. def getData_cat_and_dog(url,headers=headers):
3.     try:
4.         data = {}
5.         response = requests.get(url,headers=headers)
6.         response.encoding = response.apparent_encoding
7.         tree = etree.HTML(response.text)

```

<sup>1</sup> Trivial codes have been omitted to avoid repetition. View full codes in appendix.

```
8.         # get features
9.         flags = tree.xpath('//ul[@class="list fl"]//span[@class="flag"]/text
()')
10.        for i in range(len(flags)):
11.            part = tree.xpath(f'//ul[@class="list fl"]/li[{i+1}]/span[@class
="part"]/text()')
12.            if part:
13.                data[flags[i][: -1]] = part[0]
14.            else:
15.                data[flags[i][: -1]] = '--'
16.
17.        # get stars
18.        score_flags = tree.xpath('//ul[@class="list"]//span[@class="flag"]/t
ext()')
19.        for i in range(len(score_flags)):
20.            score = tree.xpath(f'//ul[@class="list"]/li[{i+1}]/span[@class="
part"]/i/@class')[0]
21.            if score[-1] == '0':
22.                data[score_flags[i][: -1]] = '5.0'
23.            else:
24.                data[score_flags[i][: -1]] = str(int(score[-1])/2)
25.
26.        # get picture urls
27.        pic_urls = []
28.        for url in tree.xpath('//div[@class="pet-info-txt"]//img/@src'):
29.            if url[0] == 'h':
30.                pic_urls.append(url)
31.            else:
32.                pic_urls.append(prefix + url)
33.        if pic_urls:
34.            data['图片链接'] = str(pic_urls)
35.        else:
36.            pic_urls = [ prefix + url for url in tree.xpath('//div[@class="p
et-info-t"]//img/@src')]
37.        if not pic_urls:
38.            data['图片链接'] = '--'
39.        data['图片链接'] = str(pic_urls)
40.
41.        # get knowledge
42.        char_full = []
43.        char_flags = tree.xpath('//div[@id="j-know"]/div[@class="pet-info-
t"]/text()')
44.        for i in range(len(char_flags)):
```

```
44.         char_full.append(char_flags[i])
45.         char_full.extend(tree.xpath(f'//div[@id="j-
        know"]/div[@class="pet-info-txt"][{i+1}]/*/text()'))
46.         chars = '\n'.join([char.strip() for char in char_full])
47.         if chars:
48.             data['养宠知识'] = ''.join(chars)
49.         else:
50.             data['养宠知识'] = '--'
51.
52.         # get attention
53.         .....
54.         # get species
55.         .....
56.         # get price
57.         .....
58.         # get characteristics
59.         .....
60.         # get basic_intro
61.         .....
62.         return data
63.
64.     except Exception as e:
65.         print(f"异常页面: {url}", repr(e))
```

Explicitly demonstrated in the codes above is how we re-structured the data obtained from our source by utilizing a dictionary named “data”, where each pair of fieldnames and its contents corresponds to a pair of key-value in the dictionary, respectively.

Also, whenever an exception is thrown out, the terminal prints the corresponding page link and the informative message of the error, making the processing of exception adequately convenient.

Thereafter, we conducted the following expression innovatively to directly insert the data in form of a python dictionary into MySQL:

```
1. cols = ','.join(key for key in datalist[0].keys())
2. placeholders = ','.join(f'>({key})s' for key in datalist[0].keys())
3. sql_line = f'INSERT INTO pet_data ({cols}) VALUES ({placeholders})'
```

The complete function writes bellow:

```
4. #将信息写入数据库操作
5. def insertData(datalist):
```



```
6. conn = pymysql.connect(  
7.     host='localhost',  
8.     port=3306,  
9.     user='root',  
10.    passwd='*****',  
11.    db='test',  
12.    charset='utf8')  
13.  
14.    cursor = conn.cursor()  
15.  
16.    cols = ','.join(key for key in datalist[0].keys())  
17.    placeholders = ','.join(f'%({key})s' for key in datalist[0].keys())  
18.    sql_line = f'INSERT INTO pet_data ({cols}) VALUES ({placeholders})'  
19.    for i in range(len(datalist)):  
20.        try:  
21.            cursor.execute(sql_line,datalist[i])  
22.            conn.commit()  
23.        except Exception as e:  
24.            print(i,repr(e))
```





So far, all the concrete operational functions have been accomplished. In order to further modularize the coding, we encapsulate the operational functions above, as a result of which merely two lines of function calling could suffice for the whole process of data obtainment.

```
1. #封装执行函数 1  
2. def act_data1_to_db():  
3.     urls = [prefix + getUrls(page)[k] for page in range(1,33) \  
4.     for k in range(10)] + [prefix + getUrls(33)[0]]  
5.     full_data = []  
6.     for url in urls:  
7.         full_data.append(getData_cat_and_dog(url))  
8.     insertData(full_data)  
9.  
10. #调用执行函数，完成从爬取信息到写入数据库全过程  
11. act_data1_to_db()  
12. act_data2_to_db()
```

Afterwards, complete information of all kinds of pets (dogs, cats and other categories included) have entered a table of database. Final task is to make modifications manually where negligible malposition or vacancy of data occur.

### 3.1.2 Section 2: *Service Part*

In the process of crawling articles related to pet services, some difficulties were encountered, first of all, the articles were divided into four sections, and the links between the various sections were not exactly the same, resulting in difficulties in using the same set of templates when crawling, so we established four crawler files.

 爬虫-搞笑.py	2021/11/17 15:31	PY 文件	2 KB
 爬虫-训练.py	2021/11/17 14:37	PY 文件	2 KB
 爬虫-养护.py	2021/11/17 15:27	PY 文件	2 KB
 爬虫-医疗.py	2021/11/17 15:33	PY 文件	2 KB

In the crawl of specific articles, because some articles lack some database elements (such as pictures), we used try... except syntax to ensure that the process of putting it into the database does not produce errors.

```
try:
    title=bsres0.find('h1',class_='title').text
    para1=bsres0.find('div',class_='info-box')
    para=para1.text
    img='http://www.chongqu.com.cn'+para1.find('img')['src']
    para=para.replace(' ','')
    para=para.replace('\n','')
    cursor.execute('INSERT INTO fun(title,discription,img) VALUES (%s,%s,%s)',(title,para,img))
except:
```

## 3.2 Website Construction

### 3.2.1 Home Page

#### (1) Style sheet

##### i) Briefly introduction

In order to make the website more beautiful and dynamic, we have set many exquisite styles for the website by searching, integrating and modifying some network resources. Next, we will summarize the styles in the home page and briefly introduce some styles.

##### ii) Style overview

```
1. <head>
2.     <meta charset="utf-8">
3.     <title>home</title>
4.     <link rel="stylesheet" href="/static/css/bootstrap.min.css">
5.     <link rel="stylesheet" href="/static/css/slick.css">
6.     <link rel="stylesheet" href="/static/css/magnific-popup.css">
7.     <link rel="stylesheet" href="/static/css/nice-select.css">
8.     <link rel="stylesheet" href="/static/css/linearicons.css">
9.     <link rel="stylesheet" href="/static/css/meanmenu.css">
10.    <link rel="stylesheet" href="/static/css/default.css">
11.    <link rel="stylesheet" href="/static/css/style.css">
12.    <link rel="stylesheet" href="/static/css/responsive.css">
13. </head>
```

##### iii) Details

##### @Font face:

Use the @ rules of CSS to specify online fonts for web pages, and improve the compatibility of font formats by providing their own fonts, so as to show better front-end effects.

```
1. @import url(http://fonts.useso.com/css?family=Raleway:200,500,700,800);
2. @font-face {
3.     font-family: 'icomoon';
4.     src:url('../fonts/icomoon.eot?rretjt');
5.     src:url('../fonts/icomoon.eot?#iefixrretjt') format('embedded-
   opentype'),
6.     url('../fonts/icomoon.woff?rretjt') format('woff'),
7.     url('../fonts/icomoon.ttf?rretjt') format('truetype'),
8.     url('../fonts/icomoon.svg?rretjt#icomoon') format('svg');
9.     font-weight: normal;
```



```
10. font-style: normal;
11. }
```

Some core codes:

```
1. @import url(http://fonts.useso.com/css?family=Raleway:200,500,700,800);
2. @font-face {
3.     font-family: 'icomoon';
4.     src:url('../fonts/icomoon.eot?rretjt');
5.     src:url('../fonts/icomoon.eot?#iefixrretjt') format('embedded-
        opentype'),
6.         url('../fonts/icomoon.woff?rretjt') format('woff'),
7.         url('../fonts/icomoon.ttf?rretjt') format('truetype'),
8.         url('../fonts/icomoon.svg?rretjt#icomoon') format('svg');
9.     font-weight: normal;
10.    font-style: normal;
11. }
```

Some specific font style example:

```
1. .html-leaf-header h1 {
2.     color: #fff;
3.     font-weight: 600;
4.     font-size: 2em;
5.     line-height: 1;
6.     margin-bottom: 0;
7.     font-family: "Microsoft YaHei", "宋体", "Segoe UI", "Lucida Grande", Helvetica, Arial, sans-serif, FreeSans, Arimo;
8. }
```

In order to get a better visual experience, we set up a more complex display form for the search box. It can be expanded by clicking or retracted by clicking. It mainly uses the tags in the `<span>` line and the pseudo elements of CSS: `before` and `:after` to show the complex effects.

`<span>` instance codes:

```
1. .search-wrapper .input-holder .search-icon span {
2.     width: 22px;
3.     height: 22px;
4.     display: inline-block;
5.     vertical-align: middle;
6.     position: relative;
7.     -webkit-transform: rotate(45deg);
```



```
8.     -moz-transform: rotate(45deg);
9.     transform: rotate(45deg);
10.    -webkit-transition: all .4s cubic-bezier(0.650, -
      0.600, 0.240, 1.650);
11.    -moz-transition: all .4s cubic-bezier(0.650, -0.600, 0.240, 1.650);
12.    transition: all .4s cubic-bezier(0.650, -0.600, 0.240, 1.650);
13. }
```

The slide content switching effect can make the website more dynamic and let users have a better look and experience. We use the slide plug-in based on jQuery: slick to achieve this effect.

To achieve the sliding effect, pseudo elements: before and :after are also used. In addition, :active (indicating the mouse click state), :hover (indicating the mouse hover state) and :focus (indicating the element that obtains the focus (such as form input) are also used.

Some core codes are as follows:

```
1.  .slick-track:before,
2.  .slick-track:after
3.  {
4.      display: table;
5.
6.      content: '';
7.  }
8.  .slick-track:after
9.  {
10.     clear: both;
11.  }
12. .slick-loading .slick-track
13. {
14.     visibility: hidden;
15.  }
16. .slick-slide
17. {
18.     display: none;
19.     float: left;
20.     height: 100%;
21.     min-height: 1px;
22. }
```

## (2)navigation bar settings

### i) Brief introduction

We mainly use the bootstrap submenu plug-in, which can generate multi-level submenus.

## ii) Details

Use the *href* attribute to link to the corresponding static web page to realize the function of sub menu.

Part of core codes:

```
1. <li>
2.     <a href="dog.html">宠物大全</a>
3.     <ul class="submenu">
4.         <li>
5.             <a href="dog.html">宠物狗</a>
6.         </li>
7.         <li>
8.             <a href="cat.html">宠物猫</a>
9.         </li>
10.        <li>
11.            <a href="other.html">其他宠物</a>
12.        </li>
13.    </ul>
14.</li>
```

### (3) Information expansion<sup>2</sup>

## i) Brief introduction

In this part, we still choose to show the corresponding information through the slide switching effect.

## ii) Details

In order to directly get the expansion information on the Internet, we add the *href* attribute to the *<a>* tag in each *<div>* whose class is "slide- btn" to directly connect to the corresponding website, so that users can jump from this website to other websites.

Part core details:

```
1. <div class="single-slider slider-height d-flex align-items-center" data-
   background="/static/img/slider/slider3.jpg">
2.     <div class="container">
3.         <div class="row ">
4.             <div class="col-xl-8 col-lg-10 d-flex align-items-center">
5.                 <div class="slider-content">
6.                     <h1 >养宠现状</h1>
7.                     <p >据西班牙《先锋报》网站报道，在西班牙，将近 42%的西班
                       牙家庭都养有宠物，家庭宠物的注册总数，如今已经达到 1600</p>
8.                 <div class="slide-btn">
```

<sup>2</sup> The realization of searching function will be discussed in 3.4





```
9. <a href="https://zhuanlan.zhihu.com/p/287887168" class="btn white-  
   btn">Learn More<span>→</span>  
10.                                     </a>  
11.                                     </div>  
12.                                   </div>  
13.                                   </div>  
14.                                 </div>  
15.                               </div>  
16. </div>
```

### 3.2.2 All-kinds Pages

As one of the core pages of our website, the construction of the *all-kinds* page warrant considerable concentration, which is a task combining technical application and aesthetical design.

Innovatively and definitively, we determined to adopt *bootstrap* tools (bootstrap.min.css has been downloaded and imported) for HTML design, which is a widely applied framework for HTML, CSS and JS, also used for developing responsive layout and project prioritizing mobile devices. Under bootstrap, the adjustment and design of website configuration become easier and more convenient.

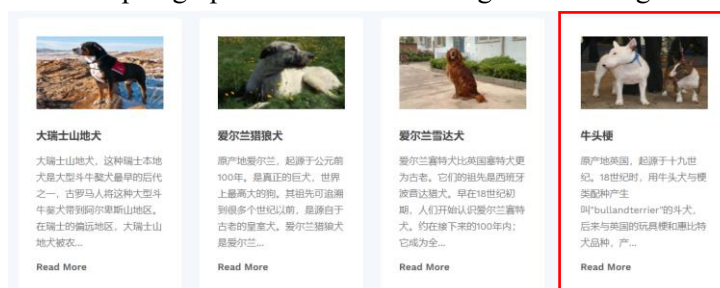
Recalling the initial designing principle that this project is to re-organize and -re-structure the information from source sites, the all-kinds page is characterized by a searching square and listed cards with concise introduction of pets on it, besides the identical header and footer of other pages.

```

1. <div class="col-lg-3">
2.   <div class="blog-wrapper blog-column" style="padding: 30px;">
3.     <div class="blog-thumb">
4.       <a href="/detail?id={{ids[1]}}">
5.         
6.       </a>
7.     </div>
8.     <div class="blog-content">
9.       <h2 class="blog-title">
10.        <a href="/detail?id={{ids[1]}}">{{names[1]}}</a>
11.      </h2>
12.      <p>{{intros[1]}}</p>
13.    </div>
14.    <div class="link-box">
15.      <a href="/detail?id={{ids[1]}}">Read More</a>
16.    </div>
17.  </div>
18. </div>

```

The above paragraph of HTML lines will generate a single card as follows in the rectangle:



In detail, each pet species has a single page demonstrating the thorough information. Apparently, the information is re-organized on the basis of the data gained from our source sites, in a way that is vivid and user-friendly.

What is worth highlight is the realization of the radar graphs illustrating the properties of each animal. The idea originated when we obtained the digital data describing each pet by characterizing their likelihood of, for example, barking or exercising, which are shown in dull texts in the original sites. To implement this, *echarts* are utilized to generate the graph in terms of given data and corresponding titles.

```
1. <script type="text/javascript">
2.   var mychart1 = echarts.init(document.getElementById('radar1'));
3.   option1 = {
4.     title: {
5.       text: '萌宠属性'
6.     },
7.     tooltip: {},
8.     legend: {
9.       bottom: 5,
10.    },
11.
12.    radar: {
13.      name: {
14.        textStyle: {
15.          color: '#fff',
16.          backgroundColor: '#999',
17.          borderRadius: 3,
18.          padding: [3, 5]
19.        }
20.      },
21.      splitNumber: 5,
22.
23.      splitArea: {
24.        areaStyle: {
25.          color: ['#ffff00', '#ffff33', '#ffff66', '#ffff99', '#ffffcc'],
26.          shadowColor: 'rgba(0, 0, 0, 0.3)',
27.          shadowBlur: 10
28.        }
29.      },
30.      indicator: [
31.        { name: '体味程度', max: 5},
32.        { name: '喜叫程度', max: 5},
33.        { name: '饲养难度', max: 5},
34.        { name: '口水程度', max: 5},
```

```

35.         { name: '掉毛程度', max: 5},
36.         { name: '运动量', max: 5},
37.         { name: '关爱需求', max: 5}
38.     ]
39. },
40.     series: [{
41.         type: 'radar',
42.         lineStyle: {
43.             width: 3,
44.             color: 'rgba(0.5,0,0,1)'
45.         },
46.         data: [
47.             {
48. value: ['{{info[15]}}', '{{info[20]}}', '{{info[21]}}', '{{info[25]}}', '{{i
49.         nfo[26]}}', '{{info[22]}}', '{{info[28]}}'],
50.             }
51.         ]
52.     }];
53.
54.     mychart1.setOption(option1);
55. </script>

```

The above lines convert the data stored in the database into explicit graphs as shown below. Note that these graphs have been rendered synchronously, meaning that specific degree of each dimension will show up dynamically when mouse is above, instead of a static picture.



One more thing to mention is the space for improving here. Provided sufficient time, it is feasible that we realized the compare system, where users are allowed to select two or more pets arbitrarily and then the website is to form a comparative radar graph showing the properties of pets chosen, respectively.

Also, it is the recommendation area of our page that is improvable, if access times are incorporated to rank the popularity of pets among people.

### 3.2.3 Service Pages

In the process of establishing the pet service page, after establishing the corresponding web page route, the first is the search box parameter problem, here with the <form> form to pass parameters, in the backend for data filtering and processing, and then the results in the database search to return the corresponding information.

```

1. <form action="service1.html" method='post'>
2.   <input type="text" placeholder="关键字或者页码（大于 0 小于 30）
   " name='page' />
3.       <button type="submit">
4.           <span class="lnr lnr-magnifier"></span>
5.       </button>
6.       <a data-animation="fadeInLeft" data-
   delay="1.2s" class="btn brand-btn" href="service1?page={{page1}}">上一页
7.       </a>
8.       <a data-animation="fadeInRight" data-
   delay="1.5s" class="btn white-btn" href="service1?page={{page2}}">下一
   页
9.       </a>
10.    </form>
11. @app.route('/service4.html', methods=['GET', 'POST'])
12. @app.route('/service4', methods=['GET', 'POST'])
13. def service4():
14.     if request.method == 'POST':
15.         page = request.form.get('page')
16.         lenght = len(page)
17.         try :
18.             low = (int(page) - 1) * 9
19.             high = (int(page) * 9)
20.             cursor.execute('SELECT * FROM fun WHERE id>{} and id<={}'.for
   mat(low, high))
21.             result = cursor.fetchall()
22.
23.             lenght = len(result)
24.             conn.commit()
25.             data = [{ for m0 in range(lenght)]
26.             for i in range(lenght):
27.                 dis = result[i][1][:35] + '...'
28.                 data[i]['title'] = result[i][0]
29.                 data[i]['dis'] = dis
30.                 data[i]['img'] = result[i][2]
31.                 data[i]['id'] = result[i][3]

```



```
32.         if int(page)>1 and int(page)<30:
33.             page1=int(page)-1
34.             page2=int(page)+1
35.         elif int(page)<=1:
36.             page1=1
37.             page2=2
38.         else:
39.             page1=29
40.             page2=30
41.     except:
42.         page='"%'+page+'"'
43.         cursor.execute('SELECT * FROM fun WHERE title LIKE {}'.forma
t(page))
44.         result=cursor.fetchall()
45.         conn.commit()
46.
47.         lenth=len(result)
48.         data=[{} for m1 in range(lenth)]
49.         for i in range(lenth):
50.             dis=result[i][1][:35]+'...'
51.             data[i]['title']=result[i][0]
52.             data[i]['dis']=dis
53.             data[i]['img']=result[i][2]
54.             data[i]['id']=result[i][3]
55.         page1=1
56.         page2=2
57.     else:
58.         try :
59.             page=request.args.get('page')
60.             low=(int(page)-1)*9
61.             high=(int(page)*9)
62.             cursor.execute('SELECT * FROM fun WHERE id>{} and id<={}'.for
mat(low,high))
63.             result=cursor.fetchall()
64.             conn.commit()
65.             data=[{} for m2 in range(9)]
66.             for i in range(9):
67.                 dis=result[i][1][:35]+'...'
68.                 data[i]['title']=result[i][0]
69.                 data[i]['dis']=dis
70.                 data[i]['img']=result[i][2]
71.                 data[i]['id']=result[i][3]
72.             if int(page)>1 and int(page)<30:
73.                 page1=int(page)-1
```





```

74.         page2=int(page)+1
75.     elif int(page)<=1:
76.         page1=1
77.         page2=2
78.     else:
79.         page1=29
80.         page2=30
81. except:
82.     page='1'
83.     low=(int(page)-1)*9
84.     high=(int(page)*9)
85.     cursor.execute('SELECT * FROM fun WHERE id>{} and id<={}'.for
mat(low,high))
86.     result=cursor.fetchall()
87.     conn.commit()
88.     data=[{} for m2 in range(9)]
89.     for i in range(9):
90.         dis=result[i][1][:35]+'...'
91.         data[i]['title']=result[i][0]
92.         data[i]['dis']=dis
93.         data[i]['img']=result[i][2]
94.         data[i]['id']=result[i][3]
95.     page1=1
96.     page2=2
97.     return render_template('service4.html',data=data,page1=page1,page2=pa
ge2)

```

Then there is the corresponding text link and page turning function, where the get parameter method is used, and the data to be transmitted is directly added to the url to achieve parameter passing.

```

1. <div class="blog-wrapper blog-column">
2.     <div class="blog-thumb">
3.         <a href="service-
details?id={{data0['id']}}&name=1">
4.             
5.         </a>
6.     </div>
7.     <div class="blog-content">
8.         <h2 class="blog-title">
9.             <a href="service-
details?id={{data0['id']}}&name=1">{{data0['title']}}</a>

```



```

10.                                     </h2>
11.                                     <p>{{data0['dis']}}</p>
12.                                     </div>
13.                                     <div class="link-box">
14.                                     <a href="service-
                                     details?id={{data0['id']}}&name=1">查看全文</a>
15.                                     </div>
16.                                     </div>

```

Finally, the recommendation of related articles is randomly selected in the corresponding database for rendering.

```

1. cursor.execute('SELECT * FROM hospital WHERE id={}'.format(id))
2.     data=cursor.fetchone()
3.     cursor.execute('SELECT * FROM hospital WHERE id >= ((SELECT MAX(i
    d) FROM hospital)-
    (SELECT MIN(id) FROM hospital)) * RAND() + (SELECT MIN(id) FROM hospital)
    LIMIT 6')
4.     otherdata=cursor.fetchall()
5.     conn.commit()
6.     return render_template('/service-
    details.html',datadata=data,name=name,otherdata=otherdata)
7. @app.route('/service-details.html',methods=['post','GET'])
8. @app.route('/service-details',methods=['post','GET'])
9. def servicedetails():
10.     request_method=request.method
11.     data=('', '', '')
12.     id=request.args.get('id')
13.     id=int(id)
14.     name=request.args.get('name')
15.     name=int(name)
16.     if name==1:
17.         cursor.execute('SELECT * FROM hospital WHERE id={}'.format(id))
18.         data=cursor.fetchone()
19.         cursor.execute('SELECT * FROM hospital WHERE id >= ((SELECT MAX(i
    d) FROM hospital)-
    (SELECT MIN(id) FROM hospital)) * RAND() + (SELECT MIN(id) FROM hospital)
    LIMIT 6')
20.         otherdata=cursor.fetchall()
21.         conn.commit()
22.         return render_template('/service-
    details.html',data=data,name=name,otherdata=otherdata)
23.     elif name==2:

```

```
24. elif name==3:  
25. elif name==4:  
26. return render_template('/service-details.html',data=data)
```

### 3.2.4 About Pages

#### (1) Brief introduction

The website construction of about page is relatively simple, which is mainly based on the navigation bar established in the previous web page, plus some <div> tags to form the division of web page layout.

```
1. <div class="page-title-area pt-210 pb-210" data-  
   background="/static/img/bg/bg8.png">  
2.     <div class="container-fluid">  
3.         <div class="row">  
4.             <div class="col-xl-12">  
5.                 <div class="page-title text-center">  
6.                     <h1>Something We Want To Say</h1>  
7.                 </div>  
8.             </div>  
9.         </div>  
10.    </div>  
11. </div>
```

#### (2) Details

Title Setting: Set some title styles and background pictures to improve the look and feel of the web page. Part core codes are quite similar to the piece shown above.

## 3.3 Data Visualization

### 3.3.1 World map visualization of pet distribution data

This project is HTML combined with Baidu map to use echarts to show the distribution of various pets (mainly dogs and cats) in the world.

This report mainly describes the implementation mode and relevant key points of the project.

#### (1) Page layout

The whole page shows the global map as much as possible, taking China as the center of the current default screen.

#### (2) Import JS file

This project is an echarts project, so you need to introduce echarts JS library, using Baidu map needs to introduce BMAP JS library. At the same time, AK (access key) is registered on Baidu development platform.

#### (3) Import library file

We can refer to the remote URL path, or you can refer to it in the form of local relative path after downloading.

The advantage of referencing in the form of web address is that the latest remote library files can be obtained synchronously, but the remote files need to be downloaded when the page is loaded;

The advantage of referencing in the form of local files is that local files can be obtained in time, but the update of library files needs to be handled separately.

The project is quoted in the form of remote website:

```
1. <script type="text/javascript"
2.   src="https://cdn.jsdelivr.net/npm/echarts@5.2.2/dist/echarts.min.js">
3. </script>
4. <script type="text/javascript"
5.   src="https://cdn.jsdelivr.net/npm/echarts@5.2.2/dist/extension/bmap.min.js">
6. </script>
```

#### (4) Introducing Baidu API

i) Register as a developer on Baidu map open platform:

<http://lbsyun.baidu.com/apiconsole/key>

ii) Create applications and obtain AK according to project requirements (according to

national policies, real name certification is required)



### iii) Introducing API

```

1. <script type="text/javascript"
2. src="https://api.map.baidu.com/api?v=2.0&ak=<Your Key Here>">
3. </script>
  
```

## (5) data model

### i) Presentation style

Mark the location of the country with data. When the mouse hovers over the mark, the country (region) name and animal name list will be displayed.

### ii) data model

The map coordinate information of each country (region) is stored in the form of object, which is convenient to search by key; Animal information is stored in the form of object list. In order to reflect the number of animals on the display, numerical information is reserved.

Storage form of national (regional) coordinate information:

```

1. geoCoordMap =
2. {
3.   '中国': [104.195397, 35.86166],
4.   .....
5.   '美国': [-95.712891, 37.09024],
6.   .....
7. }
8. data =
9. [
10.   {
  
```

```

11.         name: '中国',
12.         value: 279,
13. nameAn: '土猫,中华田园猫,中国狸花猫,黄狸猫,玳瑁色虎斑猫,大黄狗,西藏獒犬,中华田
    园犬,土狗,松狮,西施犬,京巴犬,冠毛犬,中国沙皮犬,大黄狗,北京犬,巴哥犬'
14.     },
15.     .....
16.     {
17.         name: '美国',
18.         value: 279,
19. nameAn: '加菲猫,玩偶猫,红骨猎浣熊犬,重点色短毛猫,重点色长毛猫,雪鞋猫,玩具虎猫,塞
    尔凯克卷毛猫,奇异猫,欧西猫,欧亚史烈斯猫,美国短毛猫,孟买猫,美国硬毛猫,美国刚毛
    猫,美国短尾猫,孟加拉豹猫,美国卷耳猫,褴褛猫,拉邦猫,加州闪亮猫,异国短毛猫,虎豹猫,
    虎皮猫,爪哇猫,巴厘猫,布娃娃猫,短毛波斯猫,长毛暹罗猫,布履阑珊猫,布偶猫,巴厘岛猫,
    奥西猫,玩具猎狐梗,树丛浣熊猎犬,山地犬,喜乐蒂牧羊犬,奇努克犬,Boykin Spaniel,普罗
    特猎犬,美国爱斯基摩犬,美国水猎犬,恶霸犬,斯塔福,可卡獐,黑褐猎浣熊犬,博伊金猎犬,布
    鲁克浣熊猎犬,比特犬,阿拉斯加雪橇犬,澳大利亚牧羊犬'
20.     },
21.     .....
22. ]

```

## (6) Show implementation

i) Get the ecarts diagram and draw the DOM area:

```

1. let dom = document.getElementById("container")
2. Echarts initialization
3. let myChart = echarts.init(dom)
4. Set the option property of the ecarts object. Option is the main content
   drawn by ecarts.
5. let option = {.....}
6. myChart.setOption(option)

```

ii) Data structure reconstruction

Encapsulate the national (regional) coordinates, animals and values into one object, so as to display the mark size according to the value size (symbolSize).

The result structure is:

```

1. [
2. {
3.   name: 国家区域名称,
4.   nameAn: 动物列表,
5.   value: [40.489673, 9.145000000000001, 279]
6. },
7. .....
8. ]

```

iii) Use the following method:

```
1. function convertData(data,geoCoordMap){
2.   var res = [];
3.   for (var i = 0; i < data.length; i++) {
4.     var geoCoord = geoCoordMap[data[i].name];
5.     if (geoCoord) {
6.       res.push({
7.         name: data[i].name,
8.         value: geoCoord.concat(data[i].value),
9.         nameAn: data[i].nameAn
10.      });
11.    }
12.  }
13.  return res;
14. }
```

## (7) Set coordinate center and affiliated object properties

Rewrite BMAP, set the location of China as the default display center, turn off unnecessary information such as railways, and set land, water and edge colors.

```
1. bmap: {
2.   center: [104.114129, 37.550339],
3.   mapStyle: {
4.     styleJson: [
5.       {
6.         featureType: 'water',
7.         elementType: 'all',
8.         stylers: {
9.           color: '#044161'
10.        }
11.      },
12.      {
13.        featureType: 'land',
14.        elementType: 'all',
15.        stylers: {
16.          color: '#004981'
17.        }
18.      },
19.      {
20.        featureType: 'boundary',
```





```

21.         elementType: 'geometry',

22.         stylers: {
23.             color: '#064f85'
24.         }
25.     },
26.     {
27.         featureType: 'railway',
28.         elementType: 'all',
29.         stylers: {
30.             visibility: 'off'
31.         }
32.     },
33.     .....
34. ]
35. }
36. }

```

## (8) Set rollover information

Re tooltips, format the style, and display the regional animal list in HTML.

```

1. tooltip: {
2.   trigger: 'item',
3.   formatter: function(params) {
4.     return "<div class='showbox'>所属区域: " + params.name + "<br/>动物名
      称: " + '<p class="text-wap" >' + params.data.nameAn + '</p></div>';
5.   }
6. }

```

## (9) Set option

Finally, assign (or format) relevant information to relevant attributes to obtain the final result and display it.

```

1. let option ={
2.   bmap:第 3 项内容,
3.   tooltips:第 4 项内容,
4.   series:[
5.     data:第 2 项内容
6.   ]
7. }

```

## (10) Result diagram



We can see yellow dots on the world map. These dots represent the location distribution of countries or regions, and the name of cities or regions is marked next to them. Moving the cursor to the yellow dot will display the region represented by the dot and the pet names mainly distributed in the region. And the whole world map can be dragged with the cursor to observe each position of the world map.

### 3.3.2 Data visualization of pie chart, bar chart and line chart

Compared with the visualization of world map, this part is relatively simple. We only need to use the similar codes of pie chart, column chart and line chart learned in class to fill in the data obtained from the database to get the visual chart.

```
1. import pandas as pd
2. import pyecharts.options as opts
3. from pyecharts.globals import ThemeType
4. from pyecharts.charts import Bar
5. from pyecharts.charts import Line
6. from pyecharts.charts import Pie
7. from pyecharts.charts import Grid
8. from pyecharts.charts import Page
9.
10. # 体型-饼图
11. alist = list(file['体型'])
12. xlist, ylist = get_xy(alist)
13.
14. pie1 = (
15.     Pie(init_opts=opts.InitOpts(bg_color='#FFE7BA'))
16.     .add("", [list(z) for z in zip(xlist, ylist)])
```

```
17. .set_series_opts(label_opts=opts.LabelOpts(formatter="{b}: {c}"))
18. .set_global_opts(title_opts=opts.TitleOpts("各宠物体型占比情况图")) )
```

Use pandas to read Excel data, extract the columns with specified requirements for visualization, and visualize different columns and different data with different charts:

Pycarts is used to visualize the data. Here, three kinds of charts are mainly used: histogram, pie chart and broken line chart, which correspond to pycarts Bar, pie and line diagrams in chat: Use the page component in pycarts to arrange multiple graphs, and use the draggablepagelayout mode in pycarts to drag and zoom graphs.

### 3.4 Text Retrieval

As frequently mentioned previously, the realization of searching calls for the application of text retrieval. Despite the fact that none of the group members have taken the certain parts of course about text retrieval, we figured out the method by which we conquered the plight.

Innovatively, our focus is concentrated on the tools provided by the database, where access of data derives. The fundamental function of selecting data through MySQL is to adopt the expression like:

```
1. cursor_1.execute('SELECT * FROM pet_data WHERE 宠物类别 = "犬"')
```

Speculating about further function of *select* enabled us to discover fuzzy query provided by MySQL, whose grammar appears relatively perplexing:

```
1. cursor.execute('SELECT * FROM pet_data WHERE 中文  
   名 like "%%%s%%%"'% f'{name}')
```

Namely, a formative string “%string%” as parameter to the select expression could lead to results of all items containing “string”. Making use of this characteristic, we designed the form in HTML named “userinput” to accept query from users and pass it to database for searching via Python.

```
1. cursor.execute(f'SELECT * FROM pet_data WHERE 中文名 LIKE "%{userInput}%"')
```

Above shows the expression calling fuzzy query in MySQL.

```
1. <form action="home.html" method='post'>  
2.   <div class="input-holder">  
3.     <input type="text" placeholder="Type to search" name="userinput"/>  
4.   </div>  
5. </form>
```

Above shows the structure designed in HTML to catch input contents from users.

After receiving the tuple consisting of pet names, we trace the unique ID that we granted each pet to implement accurate rendering as the searching result. The screenshot shown in the next page displays a flow of searching in our websites.

## IV. Concluding Remarks

Eventually, the whole task is accomplished after the procedures mentioned above as well as the significant contribution of joint effort. For the time being, we have endeavored to construct the website to fully fulfill the time and skills we have acquired during the course of this semester.

Recalling the designing principles we covered in the first section, it is safe and reasonable to conclude that we have managed to erect a pet website with relatively admirable explicitness and designability, compared to the source sites where all our data come from.

Academically, this project provided substantial practice about web crawlers as well as database application, data visualization and website construction, during which relevant technical proficiency have considerably progressed. It is the core principle of this course to strengthen our capability of engineering through bridging the gap between theory and practice.

To conclude, this project is a product of close cooperation and elaborate use of practical programming. For instance, over 8000 is a rough estimated number of the workload quantified by lines of codes, with a sequence of pys, js's, css's and htmls, and that each one of the group members have been concretely engaged and contributed to the accomplishment of the project.

## V. Acknowledgement

**Course:** Introduction to Engineering for Electronic Information (Class B, ICE2604)

**Instructors:** Guie Meng, Xinbing Wang

Genuine tribute and gratitude ought to be extended to all group members for their convergent collaboration, professors and all TAs in charge for their conducive instruction as well as guidance.