

Assignment3 Report

1951567 梁瀟怡

1. Introduction

At this stage, image retrieval has been widely used in search functions, and image recognition can help users quickly retrieve similar images. Instance-level image retrieval is a visual search task where the goal is, given a query image, to retrieve all images containing instances of the same object as the query image in a potentially very large image database. This project implements a system that can search based on user input pictures.

Requirement:

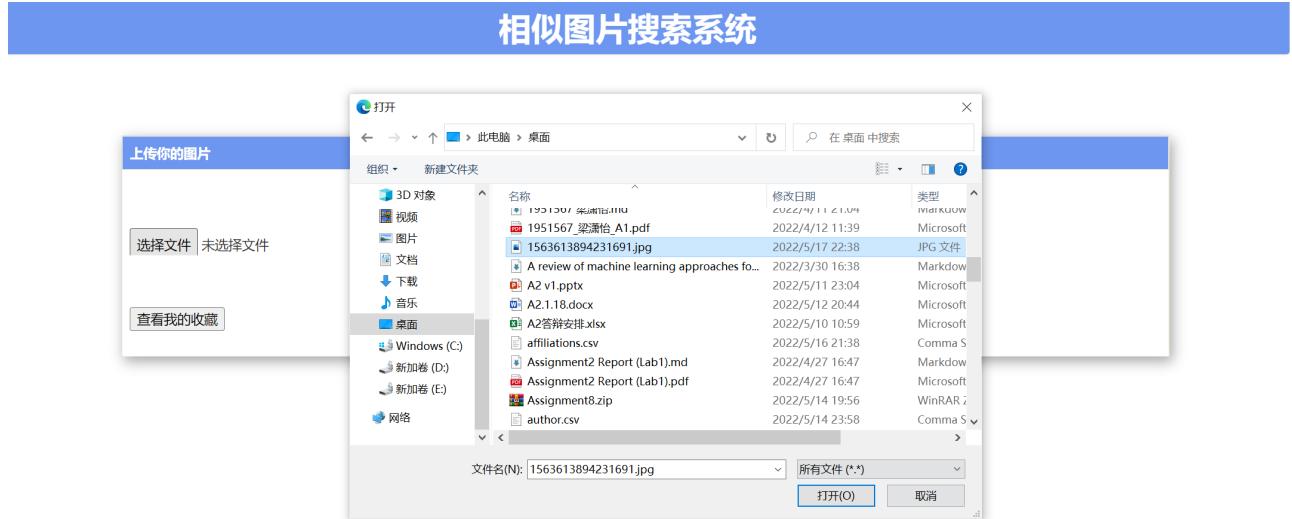
- It contains an input box to upload an image (Formulation);
- Users can preview the query image in the searching window (Formulation);
- It has a search button (Initiation);
- Provide an overview of the results (e.g. the total number of results) (Review);
- Allow changing search parameters (e.g. select certain category/tag) when reviewing results (Refinement);
- Users can take some actions, e.g. add selected images to a favorite list (Use);

Fuctions:

- Upload an image and preview it
- Search for images and display the results, including labels, previews for each result
- Filter search results based on specified tags
- Persistent Favorites

2. Formulation

In this part, the system implements a button that can be used to upload pictures, and users can click the button to upload pictures. After clicking the button, the file selection pop-up box will automatically pop up, and the user can upload any picture on the computer.



Subsequently, the user can see a thumbnail of the selected picture, which provides the user with an opportunity to confirm whether the desired picture has been selected.



The preview function is implemented as follows. Monitor the change of the upload button, by reading the uploaded image, convert it to `base64`, and use `FileReader` to convert it to `url` as the `src` attribute of the image.

```
function uploadPicture(){
    var reader = new FileReader();
    pic = document.getElementById('file').files[0];
    reader.readAsDataURL(pic);
    reader.onload = function (e) {
        document.getElementById('picImg').src=this.result;
    };
}
```

3. Initiation

The system provides a search button with the text "搜索相似图片!" to provide users with good guidance.

Click this button, the system will send a request to the backend `imgupload` interface and submit the uploaded image as a parameter. The backend processes this image to get the 9 most similar images.

After clicking the search button, the system will load the loading image, prompting the user that the system is processing the request.



The code for this part is shown in the image below.

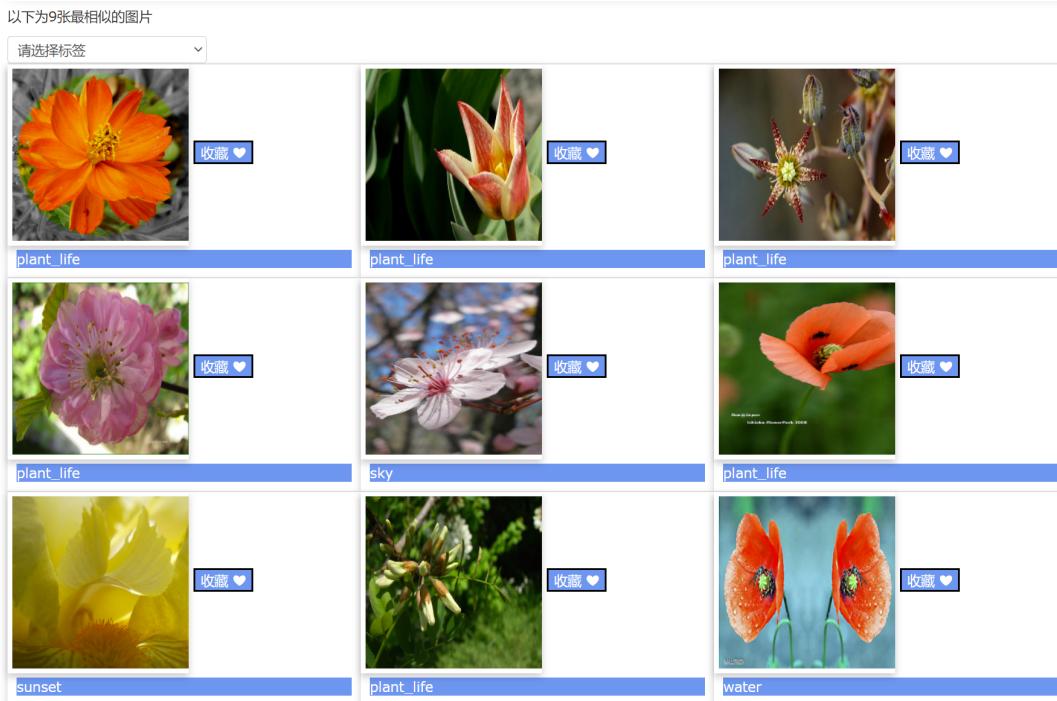
```

$( '#load' ).show();
$( "form" ).submit( function(evt){
//$( '#loader-icon' ).show();
    evt.preventDefault();
    //$( '#loader-icon' ).show();
    var formData = new FormData($(this)[0]);
    $.ajax({
        url: 'imgUpload',
        type: 'POST',
        data: formData,
        //async: false,
        cache: false,
        contentType: false,
        enctype: 'multipart/form-data',
        processData: false,
        success: function (response) {

```

4. Review

The system provides an overview of the search results, including the total number, preview of each image, tags of each image, etc.



Most of the code in this part is processed after the request result is obtained.

```

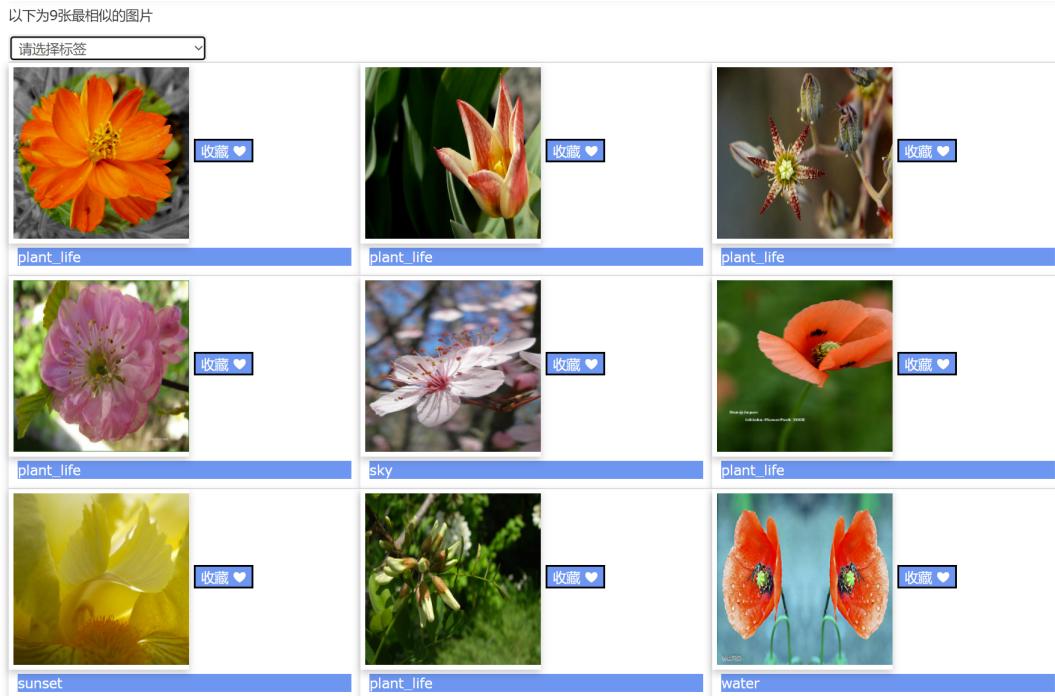
document.getElementById('tag0').innerHTML = response.tags[0];
document.getElementById('tag1').innerHTML = response.tags[1];
document.getElementById('tag2').innerHTML = response.tags[2];
document.getElementById('tag3').innerHTML = response.tags[3];
document.getElementById('tag4').innerHTML = response.tags[4];
document.getElementById('tag5').innerHTML = response.tags[5];
document.getElementById('tag6').innerHTML = response.tags[6];
document.getElementById('tag7').innerHTML = response.tags[7];
document.getElementById('tag8').innerHTML = response.tags[8];

document.getElementById("img1").src = response.image1;
document.getElementById("img2").src = response.image2;
document.getElementById("img3").src = response.image3;
document.getElementById("img4").src = response.image4;
document.getElementById("img5").src = response.image5;
document.getElementById("img6").src = response.image6;
document.getElementById("img7").src = response.image7;
document.getElementById("img8").src = response.image8;

```

5. Refinement

When displaying the search results, the system provides a drop-down selection box, and the user can select an image with a corresponding label by selecting an option in the drop-down selection box.



以下为9张最相似的图片

plant_life



收藏 ❤



收藏 ❤



收藏 ❤

plant_life



收藏 ❤

plant_life

plant_life



收藏 ❤

plant_life

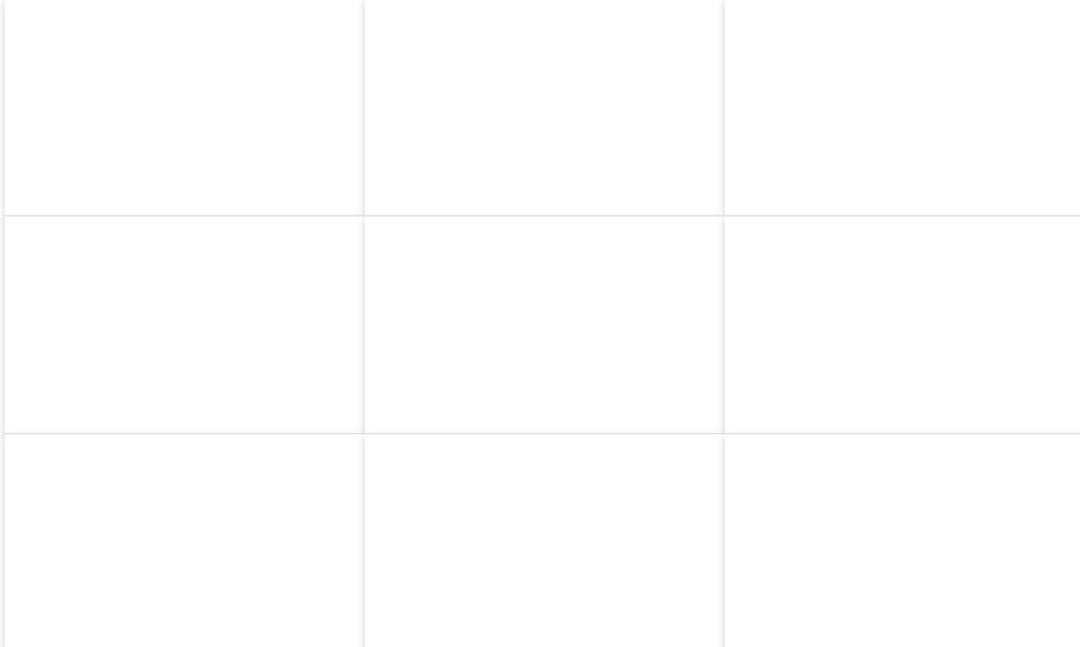


收藏 ❤

plant_life

以下为9张最相似的图片

night



以下为9张最相似的图片

sky



收藏 

sky

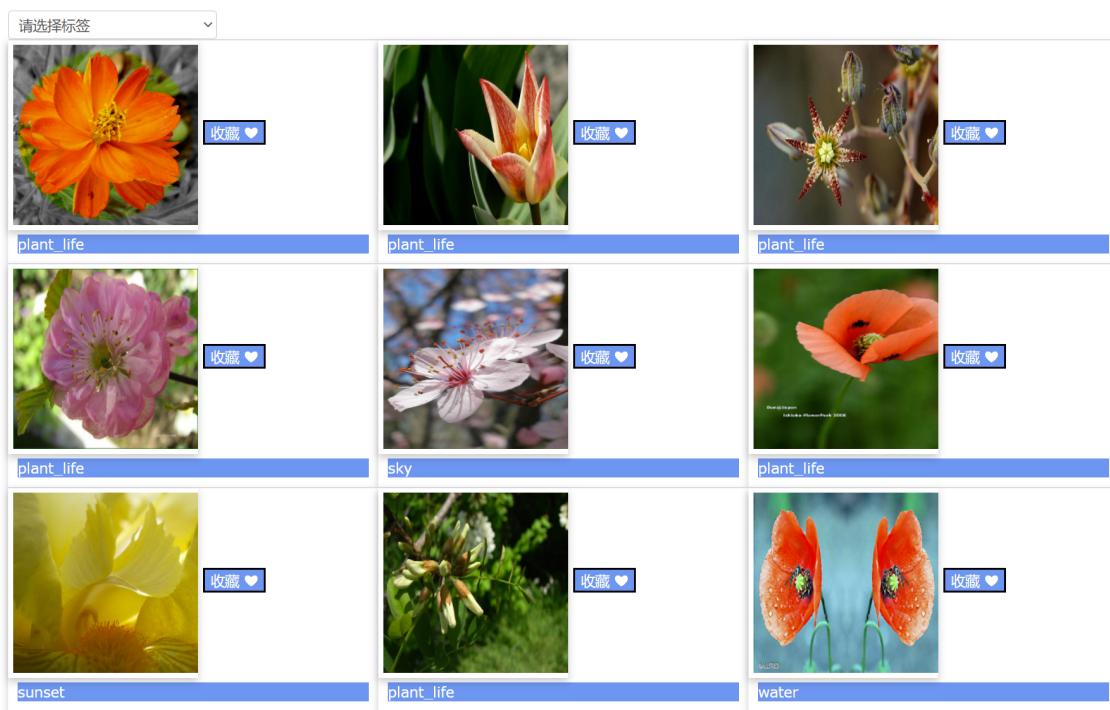
```
function handleSelect(){
    var obj = document.getElementById("select")
    var index = document.getElementById("select").options.selectedIndex
    var text = obj[index].text
    console.log(text)
    if(text !== "请选择标签"){
        for(var i=0; i<9; i++){
            var id = "tag" + i;
            var img_id = "img" + i;

            if(res.tags[i] != text){
                document.getElementById(img_id).style.visibility= "hidden"
                document.getElementById(id).style.visibility= "hidden"
                document.getElementById(img_id+"B").style.visibility= "hidden"
            }
            else{
                document.getElementById(img_id).style.visibility= "visible"
                document.getElementById(id).style.visibility= "visible"
                document.getElementById(img_id+"B").style.visibility= "visible"
            }
        }
    }
    else{
        for(var i=0; i<9; i++) {
            var id = "tag" + i;
            var img_id = "img" + i;
            document.getElementById(img_id).style.visibility= "visible"
            document.getElementById(id).style.visibility= "visible"
            document.getElementById(img_id+"B").style.visibility= "visible"
        }
    }
}
```

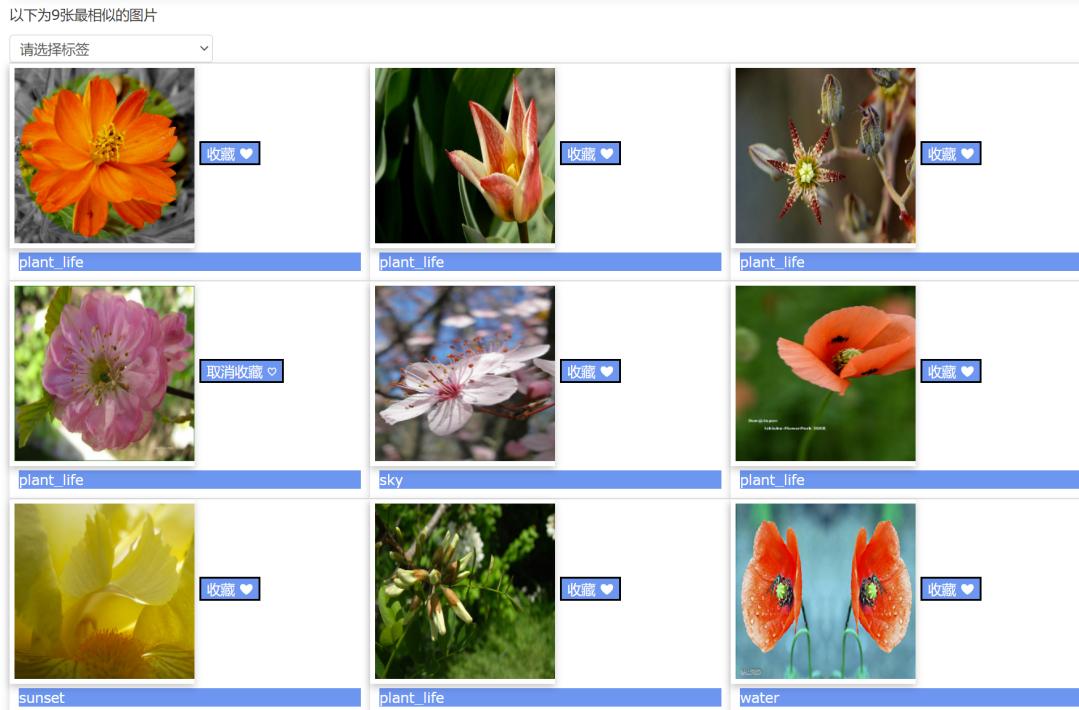
6. Use

In this part, the system provides a favorite function, the user can select a picture to favorite, cancel the favorite, and view the favorited pictures.

以下为9张最相似的图片



As shown in the picture, all the pictures are not bookmarked. Click any favorite button, you will get the following results.



As you can see, the system prompts that you have successfully saved. Click the 查看我的收藏 button to view all the pictures that have been favorited.

相似图片搜索系统

我的收藏



If you click to cancel the collection, you will also get a corresponding prompt.

100.72.93.231:5000 显示

取消收藏成功!

确定

相似图片搜索系统

我的收藏



7. Functions

7.1 Preview The Picture

```
<input type="file" name="file" id="file" required  
onchange="uploadPicture(this)" >
```

```
function uploadPicture(){  
    var reader = new FileReader();  
    pic = document.getElementById('file').files[0];  
    reader.readAsDataURL(pic);  
    reader.onload = function (e) {  
        document.getElementById('picImg').src=this.result;  
    };  
}
```

7.2 Find Result Tag

For the tag processing of the result, the first step is to group all the pictures according to the files in the tag folder, and store all the contents into a dictionary.

```
tag_dir = os.listdir('database/tags')  
tag_dict = {}  
for item in tag_dir:  
    with open('database/tags/' + item, 'r') as f:  
        res = []  
        for line in f:  
            res.append(list(line.strip('\n').split(',')))  
        # print(result)  
        for r in res:  
            tag_dict[r[0]] = item[:-4]
```

Every time a search is performed, the tag is extracted from the dictionary according to the id of the result, stored in a list, and returned to the front end.

```
if file:# and allowed_file(file.filename):  
    filename = secure_filename(file.filename)
```

```

        file.save(os.path.join(app.config['UPLOAD_FOLDER'],
filename))

        inputloc = os.path.join(app.config['UPLOAD_FOLDER'],
filename)

        recommend(inputloc, extracted_features)
        os.remove(inputloc)
        image_path = "/result"
        image_list =[os.path.join(image_path, file) for file in
os.listdir(result)

                if not file.startswith('.')]
        tags = []
        for item in image_list:
            tags.append(tag_dict[item.split("/result")[1]
[3:-4]])

        images = {
            'image0':image_list[0],
            'image1':image_list[1],
            'image2':image_list[2],
            'image3':image_list[3],
            'image4':image_list[4],
            'image5':image_list[5],
            'image6':image_list[6],
            'image7':image_list[7],
            'image8':image_list[8],
            'tags':tags
        }
        return jsonify(images)
    
```

7.3 Favorite

The collection function mainly uses the cache function of the browser. When the favorite button is clicked, the front-end stores the `src` of the image into the array in `localStorage`.

```

function favorite(item){
    var favoriteList =
JSON.parse(localStorage.getItem("favorite"));
    localStorage.removeItem("favorite")
    if(favoriteList==null){
        favoriteList = []
    }
} 
```

```
        var id = item+"B"
        if(document.getElementById(id).innerHTML === "收藏 ❤️"){
            favoriteList.push(document.getElementById(item).src);
            document.getElementById(id).innerHTML = "取消收藏
            ❤️";
            var formData = new FormData();
            formData.append("url",document.getElementById(item).src.split("res
            ult/")[1])
            console.log(formData)

            console.log(document.getElementById(item).src.split("result/"))
            $.ajax({
                url: 'addFavorite',
                type: 'POST',
                data: formData,
                cache: false,
                contentType: false,
                enctype: 'multipart/form-data',
                processData: false,
                success:function (response) {
                    console.log(response)
                });
                alert("收藏成功! ")
            }
        }
    else{
        var index =
        favoriteList.indexOf(document.getElementById(item).src);
        favoriteList.splice(index,1)
        document.getElementById(id).innerHTML = "收藏 ❤️";
        alert("取消收藏成功! ")
    }
}

localStorage.setItem("favorite",JSON.stringify(favoriteList));
}
```

Then the front end sends a request to the back-end, and the back-end copies the image from the result folder into the favorite folder, so as to ensure that when searching again, the changes to the result folder will not affect the display of favorite pictures in the "My Collection" function.

```
@app.route('/addFavorite', methods=['POST'])
def addFavorite():
    print(request.form)
    path = os.path.join("static/result", request.form["url"])
    if os.path.isfile(path):
        shutil.copy(path, "static/favorite")
    return request.url
```

Unfavorite deletes the `url` of the image from `localStorage`.

7.4 Select Tag

For the select tag, the `visibility` property in the style is used to change it dynamically.

```
function handleSelect(){
    var obj = document.getElementById("select")
    var index =
document.getElementById("select").options.selectedIndex
    var text = obj[index].text
    console.log(text)
    if(text !== "请选择标签"){
        for(var i=0; i<9; i++){
            var id = "tag" + i;
            var img_id = "img" + i;

            if(res.tags[i] != text){

document.getElementById(img_id).style.visibility= "hidden"
                document.getElementById(id).style.visibility=
"hidden"

document.getElementById(img_id+"B").style.visibility= "hidden"
            }
        else{
```

```

document.getElementById(img_id).style.visibility= "visible"
    document.getElementById(id).style.visibility=
"visible"

document.getElementById(img_id+"B").style.visibility= "visible"
}
}

}

else{
    for(var i=0; i<9; i++) {
        var id = "tag" + i;
        var img_id = "img" + i;
        document.getElementById(img_id).style.visibility=
"visible"
        document.getElementById(id).style.visibility=
"visible"

document.getElementById(img_id+"B").style.visibility= "visible"
}
}

}

```

8. How to Run

Install the requirements.

```
pip install -r requirements.txt
```

Run the server.

```
python server/image_vectorizer.py
python server/rest-server.py
```

```

loaded extracted_features
* Debugger is active!
* Debugger PIN: 116-687-048
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://100.80.108.239:5000/ (Press CTRL+C to quit)

```

Visit <https://localhost:5000> in the explorer.

