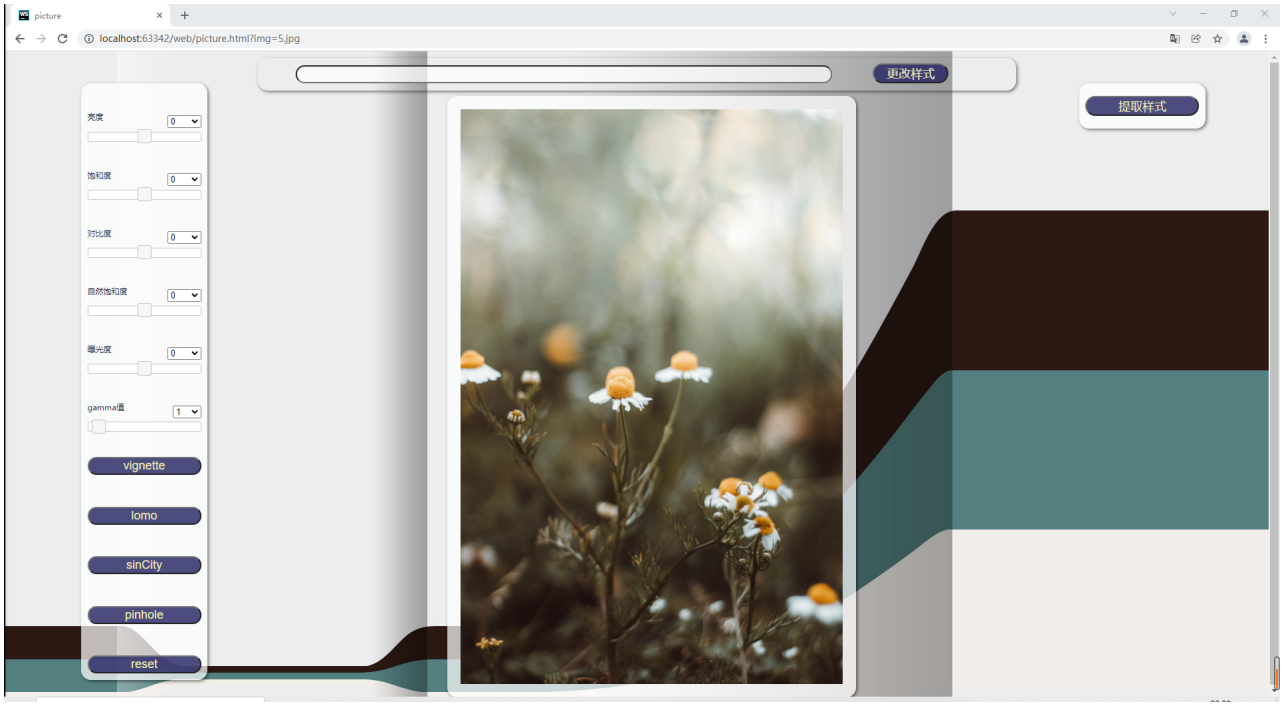
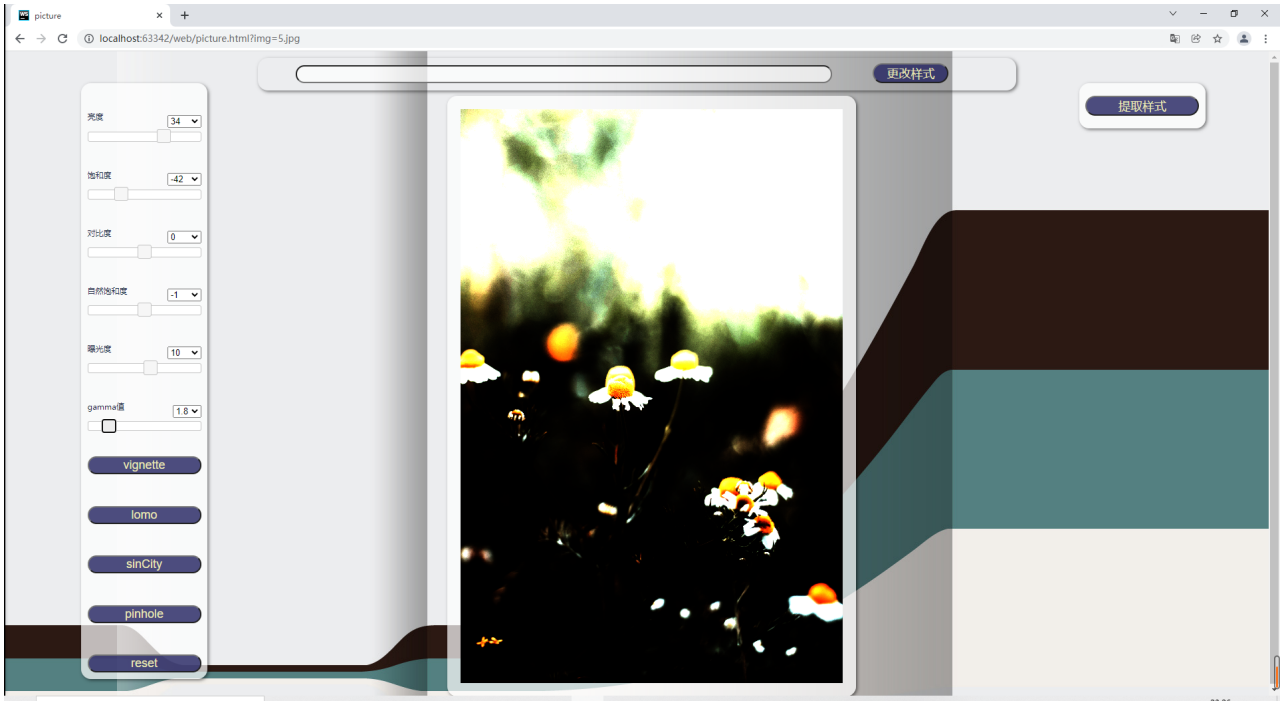
操作说明

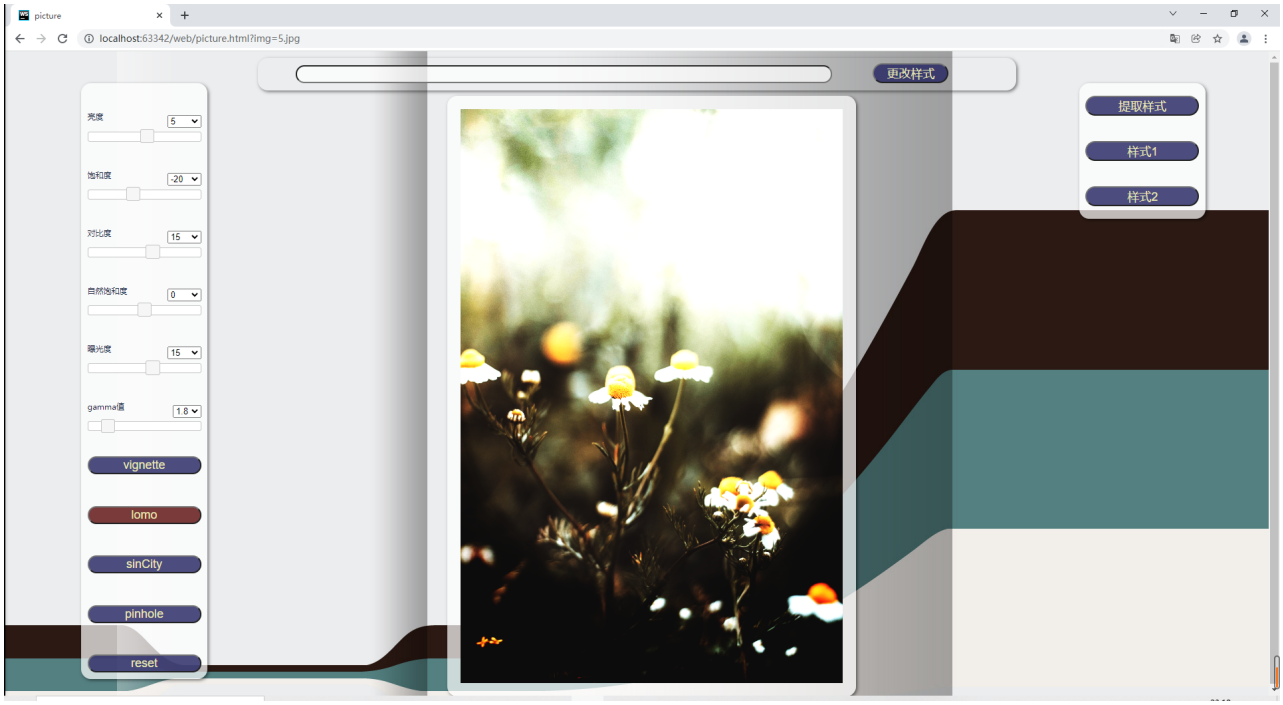
- 进入index.html，点击任意图片，即可跳转到该图片的操作页面，如下图所示



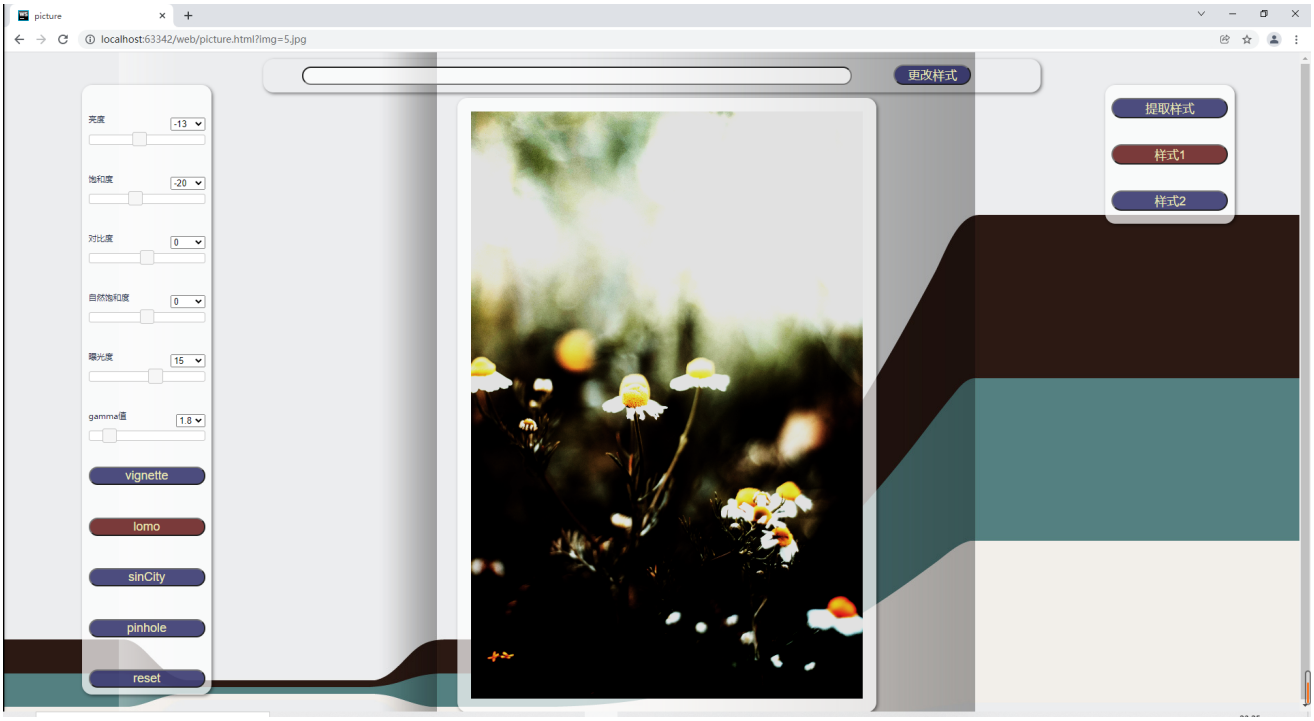
- 可以通过滑动滑块或点击选择框更改图片样貌



- 可以通过点击下方四个不同滤镜给图片添加预设好的滤镜效果



- 可以点击reset使得图片恢复原图状态
- 点击提取样式可以提取对当前图片的修改的效果，在下拉栏出现“样式x”的按钮，点击可以使得图片变为样式x的样子，并且自动复制样式码，粘贴到上方输入框再点击更改样式也可以将图片改为此样式（可以跨不同图片使用）



实现功能

- 滤镜功能，并且可以在设置滤镜的基础上再更改属性
- 手动更改图片属性，并获得所需结果
- 保存对图片属性的更改，并能在此网站其他图片应用，实现图片属性的迁移

采用的方法

- 通过调用camanJS库实现对canvas内图片各种属性的更改，调用camanJS的预设滤镜实现滤镜功能
- 使用jQuery的滑块组件，实现滑块更改的功能

关键代码说明

1. 在index.html中加入函数使得picture.htm能获取正确的图片

```
    function jump(innerHTML) {
        var a=innerHTML;
        var imgID="";
        var temp=0;
        var end=0;
        for (var i = 0; i < a.length; i++) {
            if(a.charAt(i)==='/')
                temp=i
            if (a.charAt(i)==='"')
                end=i;
        }
        imgID=a.substring(temp+1,end)
        var s='http://localhost:63342/web/picture.html?img='+imgID;
        window.location.href=s;
    }
```

2. 通过jQuery实现滑块功能，以亮度为例↓

```
        $( function() {
            var select = $( "#minbeds_1" );
            var slider = $( "<div id='slider_1' class='slider_1'></div>" ).insertAfter( select
).slider({
                min: 1,//滑块最小值
                max: 201,//滑块最大值
                range: "0",
                value: 100,//初始值
                slide: function( event, ui ) {
                    select[ 0 ].selectedIndex = ui.value - 1;//滑动时选择框数字更改
                },
                stop: function( event, ui ) {//滑动停止后选择框数字更改，并且对图片应用更改
                    select[ 0 ].selectedIndex = ui.value - 1;
                    br=ui.value - 1 -100;
                    Caman("#canvas_id",img,function(){
                        if (isVintage)//判断是否有滤镜，实现对滤镜的再次更改
                            this.vignette(400);
                        else if(isLomo)
                            this.lomo();
                        else if(isSinCity)
                            this.sinCity();
                        else if(isPinhole)
                            this.pinhole();
                        else {
                            this.gamma(ga)
                            this.vibrance(vi)
                            this.exposure(ex)
                            this.contrast(co)
                            this.saturation(st)
                        }
                        this.brightness(br).render();
                        this.revert(false);//恢复图片原始样式，但是不在浏览器中重新加载，便于下次更改
```

```
                    });
                }
            });
            slider.bind("change_bright",function (e, number) {//自定义事件，用于更改滤镜时将手动调节的参数更改为滤镜对应参数
                slider.slider( "value", number );
            })
            $( "#minbeds_1" ).bind("change_select",function(e,number){
                select[ 0 ].selectedIndex = number;//自定义事件，用于更改选择框
            })
            $( "#minbeds_1" ).on( "change", function() {
                var br=this.selectedIndex -100 + 1;
                slier_1_t=slider;
                slider.slider( "value", this.selectedIndex + 1 );
                Caman("#canvas_id",img,function(){
                    if (isVintage)
                        this.vignette(400);
                    else if(isLomo)
                        this.lomo();
                    else if(isSinCity)
                        this.sinCity();
                    else if(isPinhole)
                        this.pinhole();
                    else {
                        this.gamma(ga)
                        this.vibrance(vi)
                        this.exposure(ex)
                        this.contrast(co)
                        this.saturation(st)
                    }
                    this.gamma(ga)
                    this.vibrance(vi)
                    this.exposure(ex)
                    this.contrast(co)
                    this.saturation(st)
                    this.brightness(br).render();
                    this.revert(false);
                });

            });
        } );
```

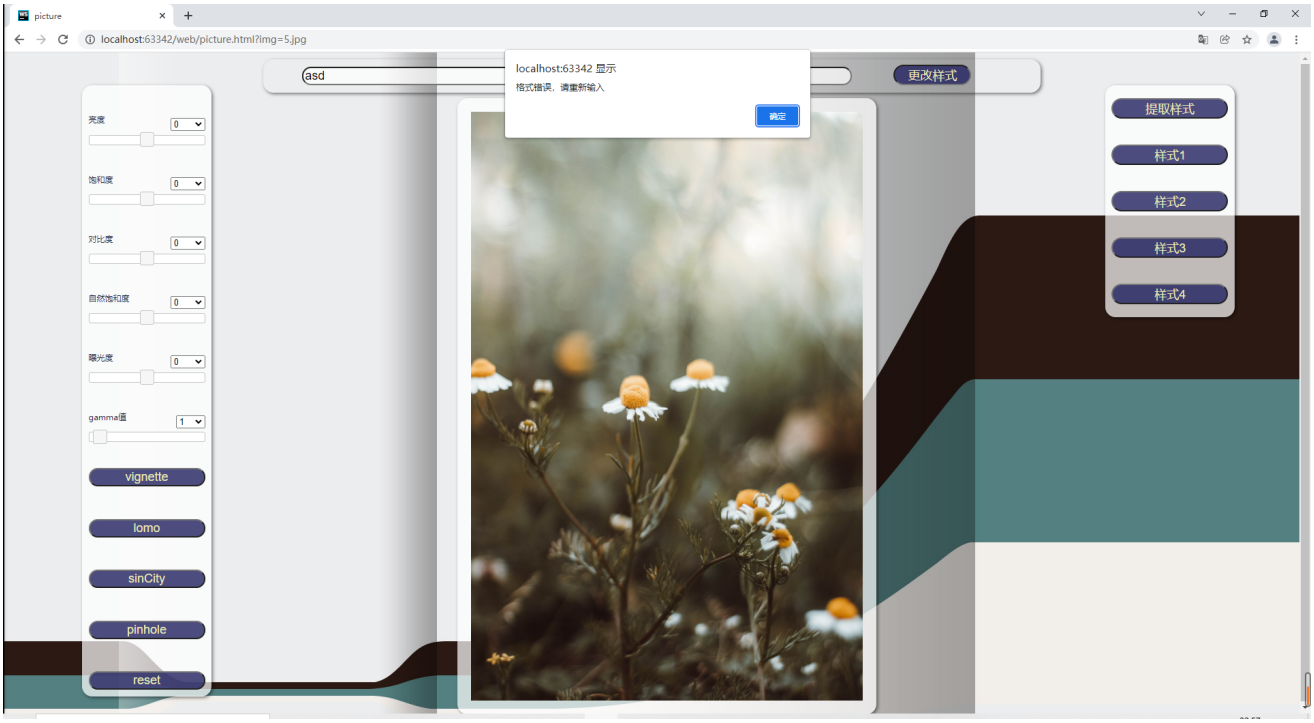
3. 提取样式的实现代码，为页面新增一个input元素

```
    function get_form() {
        let s="brightness="+br+";"
            +"saturation="+st+";"
            +"contrast="+co+";"
            +"vibrance="+vi+";"
            +"exposure="+ex+";"
            +"gamma="+ga+";"
            +"isVintage="+isVintage+";"
            +"isSinCity="+isSinCity+";"
            +"isLomo="+isLomo+";"
            +"isPinhole="+isPinhole+";"
        // s=$.parseHTML( s )
        form_num++;
```

```
        $( "<input type='button' class='form_get' onclick='form_set(value,0)'
"+"id='form_"+form_num+"'"+"value='" + "           样式"+form_num+"                "+s+"'" + "/"
+ ">" ).appendTo( $( ".form_button" ))
  }
```

4. 切换到相应样式时通过对input元素的value属性进行字符串操作获取一系列参数，应用到图片上，并用过触发自定义事件更改滑块和选择框内容，输入样式码时调用相同函数，如是输入模式，函数判断样式码是否符合规范，不符合则alert报错

```
        $( "#minbeds_1" ).trigger("change_select",br+100);
        $( "#minbeds_2" ).trigger("change_select",st+100);
        $( "#minbeds_3" ).trigger("change_select",co+100);
        $( "#minbeds_4" ).trigger("change_select",vi+100);
        $( "#minbeds_5" ).trigger("change_select",ex+100);
        $( "#minbeds_6" ).trigger("change_select",parseInt(ga*10));
        $( ".slider_1" ).trigger("change_bright",br+100);
        $( ".slider_2" ).trigger("change_saturation",st+100);
        $( ".slider_3" ).trigger("change_contrast",co+100);
        $( ".slider_4" ).trigger("change_vibrance",vi+100);
        $( ".slider_5" ).trigger("change_exposure",ex+100);
        $( ".slider_6" ).trigger("change_gamma",parseInt(ga*10));
```



亮点与缺陷

亮点

1. 实现对图片的更改操作一体化，不论在哪里更改图片属性，手动的滑块和选择框都能变为对应的参数
2. 实现了网站内多个图片的图像更改属性迁移
3. 可以记录不同的样式并对比回看

缺陷

1. 由于滤镜的相关参数是查询了camanJS的源码获取得到，但是源码内部分属性不全，使得无法通过手动调参获得与滤镜相同的效果
2. 可以调节的参数与camanJS的示例对应起来较少，因为再添加调节的属性已没有技术上的难点，都是重复性的工作，所以不在这里多加

一点bug修改记录

1. 图片长宽比不同加载过来的时候页面会变得很怪异，于是根据长宽比在加载时添加不同的style属性，保证页面的工整
2. 因为要实现滑块和选择框与滤镜更改等的对应关系，不得不在其函数外调整，但是调整半天都不可行，由于其元素加载周期过于复杂，之后遂干脆自定义更改事件。
3. 由于每次对图片的操作都要返回一次图片初始状态，所以滑块拖动的时候会发生闪烁，最后看jQuery的api后将更改图片的代码放在滑块的stop事件发生时。