

Lee sunmoon

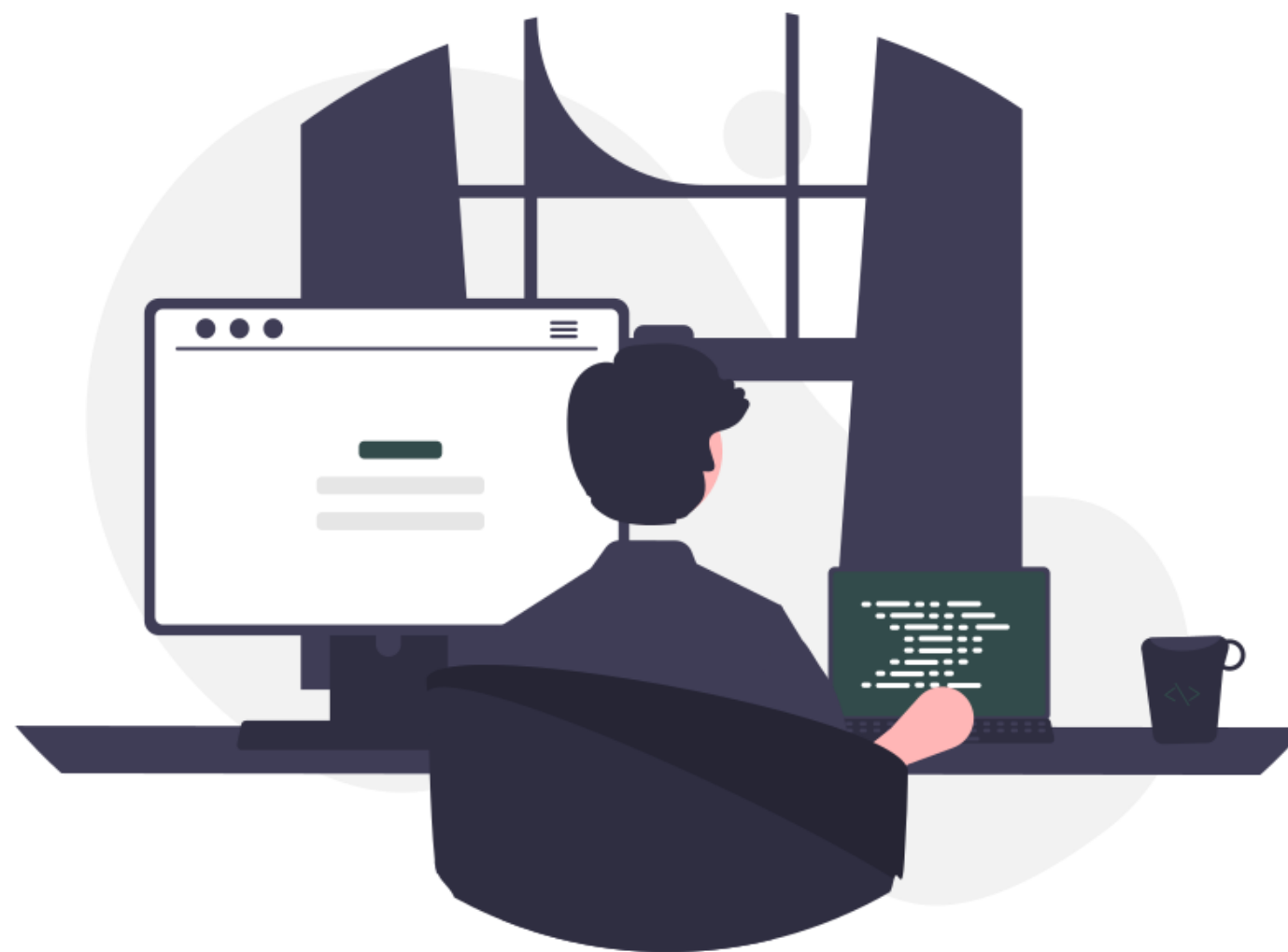
MyDiary

Portfolio



# CONTENTS

- 타임라인 p.3
- 주제 선정 p.4
- 컨셉 설정 p.5
- UI 설계 p.6~7
- 구현 작업 p.8~15
- 마무리 p.16~17



## • 타임라인



- 주제 선정

사람은 망각의 동물이라고 합니다. 그래서 사람은 기록을 남깁니다. 기록을 남기면서 개인적인 목표를 설정하고 진행을 추적하는 데 도움을 줄 수 있고 개인적인 정보 (예를 들어 일정, 할 일 목록, 지출 내역 등)을 기록하는 데 사용할 수도 있습니다. 또 중요한 순간이나 이벤트를 기록하여 추후 기억하고 되새길 수 있지요.

"기록되지 않는 것은 기억되지 않는다"는 말처럼 기록에 도움을 주는 웹 사이트를 만들자는 생각을 했습니다. 언제 어디서나 자신의 기록을 남길 수 있는 나만의 다이어리를 만들었습니다.

- 컨셉 설정 Color - System



Point Color

C8% M0% Y0% K5%

R222 G242 B242

#DEF2F2



Sub Color 1

C33% M0% Y0% K71%

R50 G75 B75

#324B4B



Sub Color 2

C33% M1% Y0% K29%

R122 G181 B182

#7AB5B6

## • UI 설계 - 와이어프레임

Sunmoon's Dairy

번호	제목	날짜
1	빗방울 하나가 - 강은교	23.08.28
2		
3		
4		
5		
6		
7		
8		
9		
10		

2023년 8월 29일 화요일
날씨 비내림

<디폴트 이미지>

빗방울 하나가 - 강은교

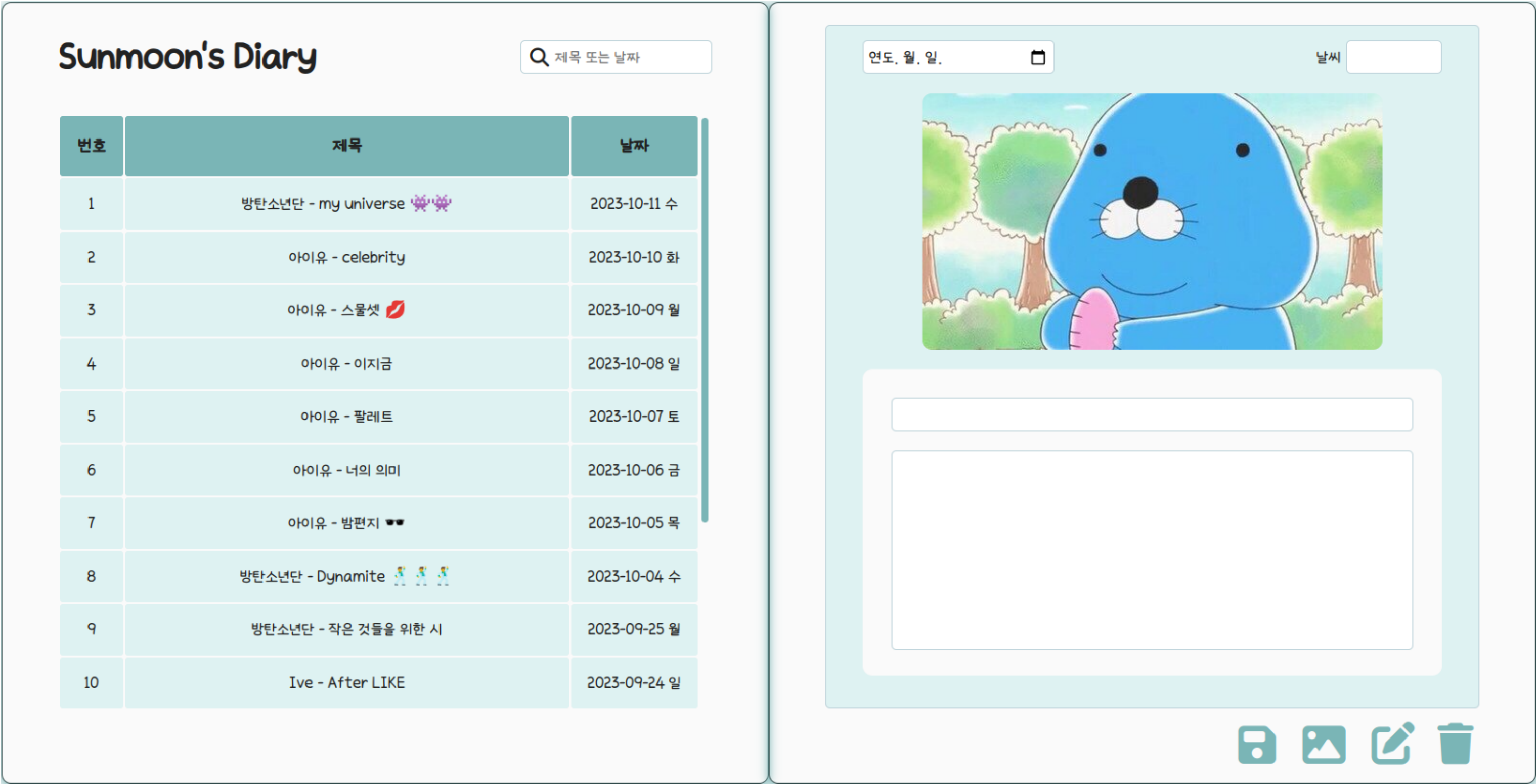
무엇인가 창문을 두드린다  
놀라서 소리나는 쪽을 바라본다  
빗방울 하나가 서 있다가 쪼르르 떨어져 내린다  
  
우리는 언제나 두드리고 싶은 것이 있다  
그것이 창이든, 어둠이든  
또는 별이든

사람들이 일반적으로 사용하는 다이어리의 모습을 최대한 구현하려고 했습니다.

왼쪽에는 목록, 오른쪽에는 작성 내용이 들어가는 자리입니다. 일기 내용뿐만 아니라 사진도 같이 넣을 수 있게 했습니다.

아이콘을 활용해 직관적으로 작동 방식을 보여주려 했습니다.

• UI 설계 - 완성된 화면



- 구현 작업

- HTML
- CSS
- JavaScript
- python
- django
- MariaDB





- 구현 작업

django views에서 Create, Read를 구현한 코드

```
@csrf_exempt
def create( request ):
    if bool( request.FILES.get( 'photo', False ) ) :
        MyDiary.objects.create( get_date = request.POST["date"], title = request.POST["title"], detail = request.POST["detail"], photo = request.FILES["photo"], wheather = request.POST["wheather"] )
    else:
        MyDiary.objects.create( get_date = request.POST["date"], title = request.POST["title"], detail = request.POST["detail"], wheather = request.POST["wheather"] )
    return redirect('/')

```

HTML에서 POST 방식으로 받은 데이터를 쿼리셋을 사용하여 DB에 정보를 전달하게 했습니다.

```
@csrf_exempt
def read( request, id ):
    try:
        diary = MyDiary.objects.get( id = id )
        values = MyDiary.objects.order_by( "-get_date" )[:50]
    except MyDiary.DoesNotExist:
        raise Http404( "일기가 존재하지 않습니다!" )
    return render( request, "diary/read.html", { "diary": diary , "values" : values } )

```

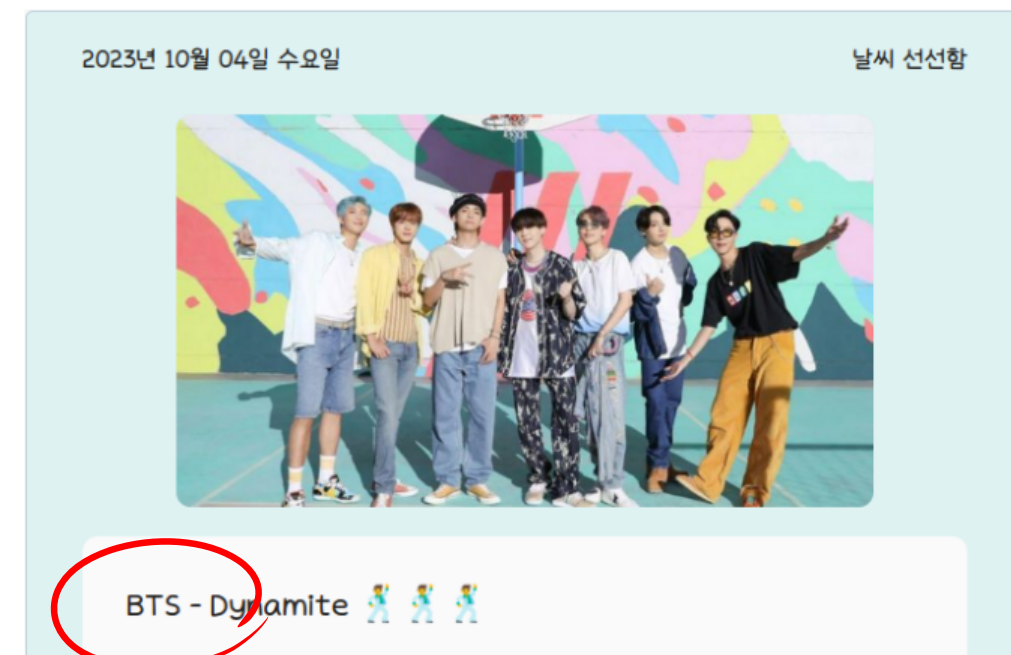
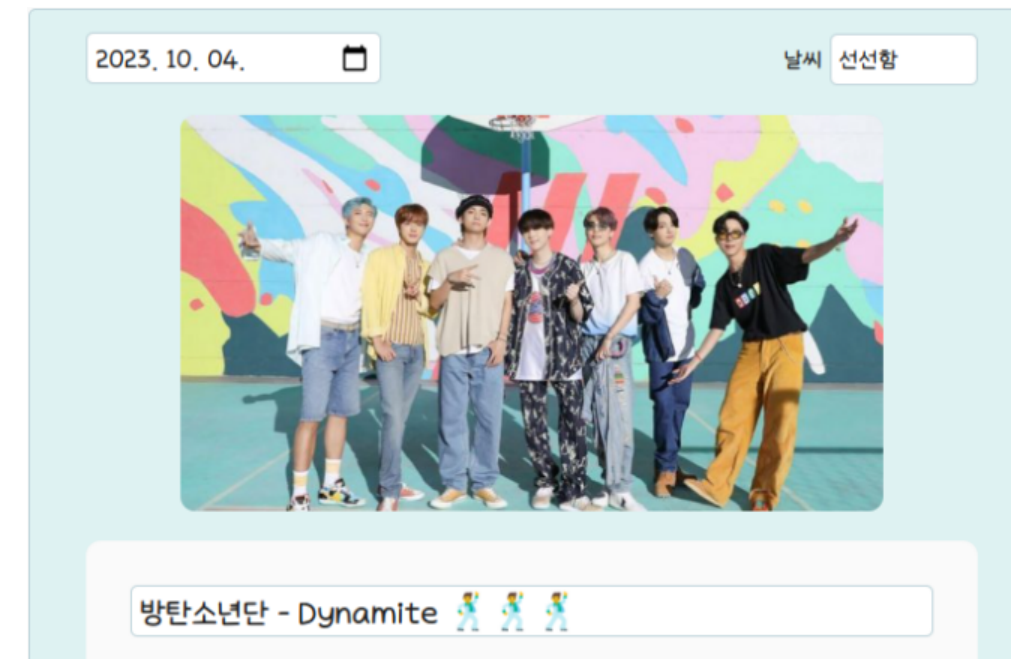
HTML에서 id를 가져와서 어떤 정보를 가져올지 특정할 수 있게 했습니다. 일기가 없을 경우에는 Http404 에러가 나오게 했습니다.

## • 구현 작업

django views에서 Update, Delete를 구현한 코드

```
@csrf_exempt
def update( request, id ):
    diary = MyDiary.objects.get( id = id )
    values = MyDiary.objects.order_by( "-get_date" )[:50]
    if request.method == "POST":
        if bool( request.FILES.get( 'photo', False ) ) :
            diary.get_date = request.POST["date"]
            diary.title = request.POST["title"]
            diary.detail = request.POST["detail"]
            diary.photo.delete()
            diary.photo = request.FILES["photo"]
            diary.wheather = request.POST["wheather"]
            diary.save()
        else:
            diary.get_date = request.POST["date"]
            diary.title = request.POST["title"]
            diary.detail = request.POST["detail"]
            diary.wheather = request.POST["wheather"]
            diary.save()
        print(request.POST.get('date', ''))
    return render( request, "diary/update.html", { "diary": diary , "values": values } )
```

HTML에서 id 값을 전달 받아 어떤 다이어리의 내용을 수정할지 가져와서 쿼리셋을 활용해 가져옵니다. 수정한 내용으로 덮어쓰우고 저장하면 내용이 업데이트 됩니다.



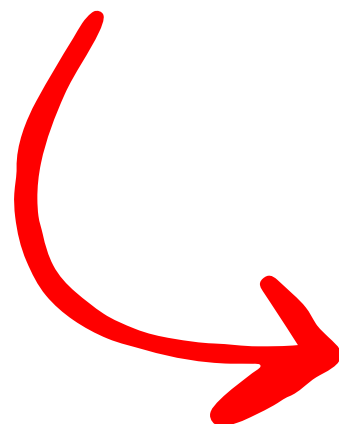
## • 구현 작업

django views에서 Update, Delete를 구현한 코드

```
@csrf_exempt
def delete( request ):
    MyDiary.objects.filter( id = request.POST["id"] ).delete()
    return redirect( "/" )
```

7	아이유 - 밤편지 🌙🌙	2023-10-05 목
8	BTS - Dynamite 🧑🧑🧑	2023-10-04 수
9	딘 - 넘어와	2023-09-26 화
10	방탄소년단 - 작은 것들을 위한 시	2023-09-25 월

HTML에서 id 값을 전달 받아 어떤 다이어리의 내용을 삭제할지 가져오고 쿼리셋을 활용해 특정하여 DB에서 지울 수 있도록 합니다.



7	아이유 - 밤편지 🌙🌙	2023-10-05 목
8	BTS - Dynamite 🧑🧑🧑	2023-10-04 수
9	방탄소년단 - 작은 것들을 위한 시	2023-09-25 월
10	Ive - After LIKE	2023-09-24 일

## • 구현 작업

django ORM을 활용하여 MariaDB와 연동

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'c18st21',
        'USER': 'c18st21',
        'OPTIONS': {"charset": "utf8mb4"},
    }
}
```

settings에서 db와 유저를 추가합니다.

```
from django.db import models

class MyDiary( models.Model ):
    get_date = models.DateField( max_length = 15 )
    title = models.CharField( max_length = 30 )
    detail = models.TextField()
    photo = models.ImageField( upload_to = "images/", null=True , default="null" )
    wheather = models.CharField( max_length = 15 , null=True , default="null" )
```

django model에서 ORM을 써서 DB  
에 테이블을 생성하는 명령문을 넣습니다.

```
+-----+
| auth_group |
| auth_group_permissions |
| auth_permission |
| auth_user |
| auth_user_groups |
| auth_user_user_permissions |
| diary_mydiary |
| django_admin_log |
| django_content_type |
| django_migrations |
| django_session |
| phones_information |
| userTable |
+-----+
```

생성된 테이블 diary\_mydiary

## • 구현 작업

### Template 문법을 활용하여 mysql과 연동

```
{% for v in values %}
<ul class="listItem">
  <li><a href="{% url 'diary:read' v.id %}#rightDiary">{{ forloop.counter }}</a></li>
  <li><a href="{% url 'diary:read' v.id %}#rightDiary" class="listTitle">{{ v.title }}</a></li>
  <li><a href="{% url 'diary:read' v.id %}#rightDiary" class="listDate">{{ v.get_date|date:'Y-m-d D'|slice:'0:12' }}</a></li>
</ul>
{% endfor %}
```

view에서 template으로 보낸 객체를 values로 받아서 for문으로 나열했습니다.

forloop.counter로 가져온 객체 수를 번호로 표현했습니다. title은 title안의 제목을 가져오고 date는 django date filter를 활용해 날짜를 표기했습니다. 또 slice를 활용해 '월요일'에서 '요일'을 빼 '월'만 표시했습니다.

번호	제목	날짜
1	방탄소년단 - my universe 🌸🌸	2023-10-11 수
2	아이유 - celebrity	2023-10-10 화
3	아이유 - 스물셋 🍷	2023-10-09 월
4	아이유 - 이지금	2023-10-08 일
5	아이유 - 팔레트	2023-10-07 토
6	아이유 - 너의 의미	2023-10-06 금
7	아이유 - 밤편지 📖	2023-10-05 목
8	방탄소년단 - Dynamite 🧨🧨🧨	2023-10-04 수

## • 구현 작업

### JS로 Search 기능 구현

```
//search 기능
let diaryList = document.getElementsByClassName( "ListItem" );
document.getElementById( 'searchText' ).addEventListener( "keyup", () => {
  let searchVal = document.getElementById( 'searchText' ).value;
  for ( let i = 0; i < diaryList.length; i++ ) {
    if (diaryList[i].querySelector( '.listTitle' ).innerText.includes( searchVal ) || diaryList[i].querySelector( '.listDate' ).innerText.includes( searchVal ) ) {
      diaryList[i].style.display = '';
    } else {
      diaryList[i].style.display = 'none';
    }
  }
});
```

DB에서 가져온 일기 목록에 title과 date의 innerText와 검색창의 value를 비교해서 포함되어 있는 글자를 전부 찾습니다.

일치하는 값이 있다면 display하고, 없으면 display='none'으로 표시하지 않습니다.

#### Sunmoon's Diary

번호	제목	날짜
2	아이유 - celebrity	2023-10-10 화
3	아이유 - 스물셋 🍷	2023-10-09 월
4	아이유 - 이지금	2023-10-08 일
5	아이유 - 팔레트	2023-10-07 토
6	아이유 - 너의 의미	2023-10-06 금
7	아이유 - 밤편지 📧	2023-10-05 목



## • 구현 작업

### JS로 이미지 미리보기 구현

```
//이미지 미리보기
document.addEventListener( "DOMContentLoaded", function () {
  const imageInput = document.getElementById( "inputImage" );
  const imagePreview = document.getElementById( "imageUpload" );

  imageInput.addEventListener( "change", function () {
    const selectedImage = imageInput.files[0];
    if ( selectedImage ) {
      const reader = new FileReader();

      reader.onload = function () {
        imagePreview.src = reader.result;
      };
      console.log( reader.result );
      reader.readAsDataURL( selectedImage );
    } else {
      imagePreview.src = ""; // 이미지가 선택되지 않은 경우 미리보기 초기화
    }
  });
});
```



이미지가 업로드 되면 HTML내의 img태그 경로를 생성합니다. FileReader가 이미지를 생성합니다.

## • 마무리

처음으로 장고라는 파이썬 웹 프레임워크를 이용해서 MVT 모델을 활용해 웹 개발을 해보았습니다. 처음에 CORS 에러를 해결하는 데 시간이 오래 걸렸지만, 이 에러 덕분에 웹 개발에 대한 많은 것을 다시 알게 되었습니다.

이전까지는 웹에 어떤 것을 입력해도 저장을 할 수 없는 프론트엔드만 개발했지만, 이렇게 직접 DB에 정보를 저장하고 불러오는 개발을 해보니까 백엔드가 얼마나 매력적인지 알게 되었습니다. 하지만 두가지 작업을 같이 하니 시간이 더 오래 걸렸습니다. 다음에는 개발 속도를 더 높여서 작업하도록 하겠습니다.



감사합니다!



다음 포트폴리오에서 만나요!



MyDiary 바로가기