

CRUD

〈CRUD〉

CRUD 란
데이터를 처리를 위해 필요한 기능으로

Create :: 데이터 생성
Read :: 데이터 조회(읽기)
Update :: 데이터 수정
Delete :: 데이터 삭제

를 통틀어 칭하는 말입니다.

〈CRUD〉

Model

데이터베이스 (DB) 를 담당하는 model은 어떤 정보를 받을지 담당합니다!

이는 excel 처럼 표의 형태로 정리하여 DB에 추가하는데요

DB를 다루기 위해 학교 수업에서는 SQL을 주로 사용하지만,
Django는 DB 관리에 있어 쉽게 연결해주는 ORM을 사용합니다.

ORM이란

Object-Relational Mapping의 약자입니다 ~

주의해야할 점은

model은 DB를 다루는 것이지 model == DB는 아닙니다!

〈CRUD〉

Template

**template은 사용자에게 보여지는 부분으로,
저희가 Django 실습을 했을 때 templates 폴더 안에
html 파일을 작성했던 것 기억하시죠 ??**

**이와 같이 html 파일이 template을 담당합니다.
Django 템플릿 시스템 문법에 맞게 작성하면
데이터를 사용자의 화면에 불러올 수 있게 됩니다!
주로 템플릿 변수, 템플릿 필터, 템플릿 코멘트, 템플릿 태그로 이루어져 있습니다.**

〈CRUD〉

View

**View는 웹에서 요청을 받거나 전달받은 데이터를 가공하여
Template에 띄우도록 하는 것입니다!**

데이터가

어떤 상황에서

어떤 식으로

처리되는지 이에 대한 함수를 모아둔 것이라고 생각하면 되어요.

작성한 view와 template을 연결하기 위해서는
url을 설계하여 연결합니다.

〈CRUD〉

오늘 실습에 주로 사용할 template 을 알아볼게요

index.html :: 런서버를 하면 바로 보이는 페이지

create.html :: 글을 작성하는 페이지 (C에 해당)

read.html :: 작성한 모든 글을 모아보는 페이지 (R에 해당)

detail.html :: 글의 세부 내용을 보여주는 페이지 (R에 해당)

〈환경설정〉

1. 바탕화면에 crud1 폴더를 생성 후 Vscode에서 열어주세요

2. Vscode 터미널을 열어
가상환경을 생성해주세요

Windows :: **python -m venv venv**

Mac :: **python3 -m venv venv**

3. 가상환경을 실행해주세요

Windows :: **source venv/Scripts/activate**

Mac :: **source venv/bin/activate**

4. Django를 설치해주세요.

Windows & Mac :: **pip install django**

〈환경설정〉

5. ‘crud1project’ 프로젝트를 생성해주세요.

Windows & Mac :: django-admin startproject crud1project .

6. ‘crudapp’ 앱을 생성해주세요

Windows :: python manage.py startapp crudapp

Mac :: python3 manage.py startapp crudapp



```
vounabeen@YounaBeenui-MacBookAir ~ ~/Desktop/crud1 python3 -m venv venv
● youngbeen@YoungBeenui-MacBookAir ~ ~/Desktop/crud1 source venv/bin/activate
● (venv) youngbeen@YoungBeenui-MacBookAir ~ ~/Desktop/crud1 pip install django
Collecting django
  Using cached Django-4.2.1-py3-none-any.whl (8.0 MB)
Collecting asgiref<4,>=3.6.0
  Using cached asgiref-3.6.0-py3-none-any.whl (23 kB)
Collecting sqlparse>=0.3.1
  Using cached sqlparse-0.4.4-py3-none-any.whl (41 kB)
Installing collected packages: sqlparse, asgiref, django
Successfully installed asgiref-3.6.0 django-4.2.1 sqlparse-0.4.4

[notice] A new release of pip available: 22.2.1 → 23.1.2
[notice] To update, run: pip install --upgrade pip
● (venv) youngbeen@YoungBeenui-MacBookAir ~ ~/Desktop/crud1 django-admin startproject crud1project .
● (venv) youngbeen@YoungBeenui-MacBookAir ~ ~/Desktop/crud1 python3 manage.py startapp crudapp
```

이렇게
되어야 해요

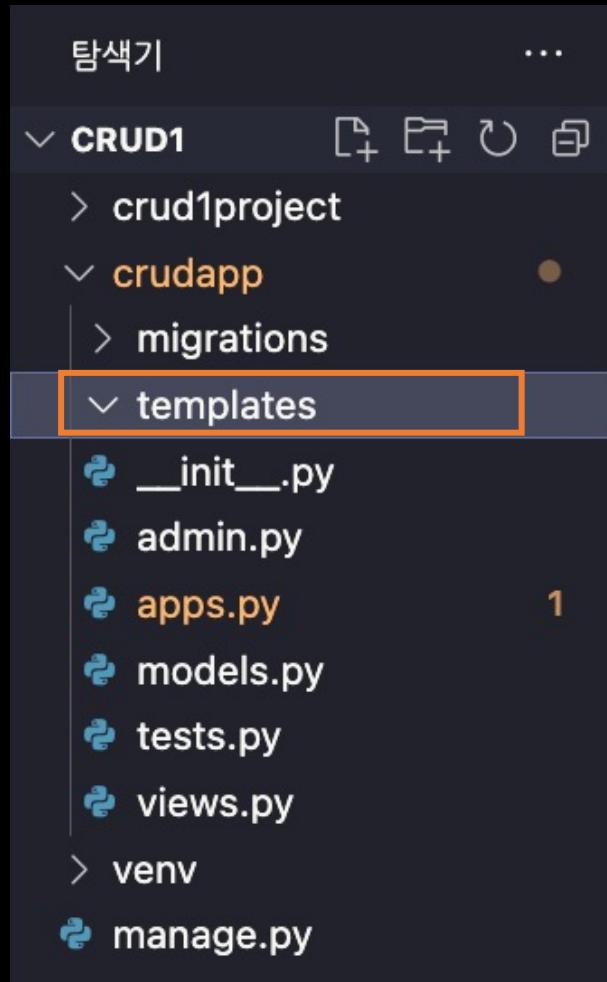
〈환경설정〉

7. ‘crud1project’ 의 settings.py 에 생성한 ‘crudapp’ 정의하기

```
crud1project > 📄 settings.py > ...  
...  
33     INSTALLED_APPS = [  
34         'django.contrib.admin',  
35         'django.contrib.auth',  
36         'django.contrib.contenttypes',  
37         'django.contrib.sessions',  
38         'django.contrib.messages',  
39         'django.contrib.staticfiles',  
40         'crudapp.apps.CrudappConfig'  
41     ]
```

〈환경설정〉

8. 'crudapp' 속에 templates 폴더 생성



9. 'crud1project' 프로젝트 폴더의 settings.py에서 secret key를 복사해서

```
crud1project > settings.py > ...
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = 'django-insecure-th=@39-ch*8cad)3n(!u0kzqnu*4tfc3by4w6+!l$5bvffcsn$'
24
```

manage.py 와 같은 위치에
secrets.json 파일을 만들고
이렇게 작성해주세요.

```
{} secrets.json > ...
1 {
2   "SECRET_KEY" : "django-insecure-th=@39-ch*8cad)3n(!u0kzqnu*4tfc3by4w6+!l$5bvffcsn$"
3 }
```

```
{
  "SECRET_KEY" : "settings.py의 secret key 값"
}
```

〈환경설정〉

10. ‘crud1project’ 프로젝트 폴더의 settings.py에 import 해주고

```
crud1project > ✎ settings.py > ...
13  from pathlib import Path
14  import os, json
15  from django.core.exceptions import ImproperlyConfigured
16
```

〈환경설정〉

11. secrets.json에서 secret key를 불러오기 위한 코드를 작성해줄게요.

crud1project > 📄 settings.py > ...

```
24 # SECURITY WARNING: keep the secret key used in production secret!
25 secret_file = os.path.join(BASE_DIR, 'secrets.json')
26
27 with open(secret_file) as f:
28     secrets = json.loads(f.read())
29
30 def get_secret(setting, secrets = secrets):
31     try:
32         return secrets[setting]
33     except KeyError:
34         error_msg = "Set the {} enviroment variable".format(setting)
35         raise ImproperlyConfigured(error_msg)
36
37 SECRET_KEY = get_secret("SECRET_KEY")
38
```

〈환경설정〉

12. Google에 gitignore.io를 검색하고 Python, Django를 선택하여 생성한 문서를 복사한 후

› **gitignore.io**

자신의 프로젝트에 꼭 맞는 .gitignore 파일을 만드세요

Python ×

Django ×

생성

```
# Created by https://www.toptal.com/developers/gitignore/api/python,django
# Edit at https://www.toptal.com/developers/gitignore?templates=python,django

### Django ####
*.log
*.pot
*.pyc
__pycache__/
local_settings.py
db.sqlite3
db.sqlite3-journal
media

# If your build process includes running collectstatic, then you probably don't need or want to include staticfiles/
# in your Git repository. Update and uncomment the following line accordingly.
# <django-project-name>/staticfiles/

### Django-Python Stack ####
# Byte-compiled / optimized / DLL files
*.py[co]d
*Py.class

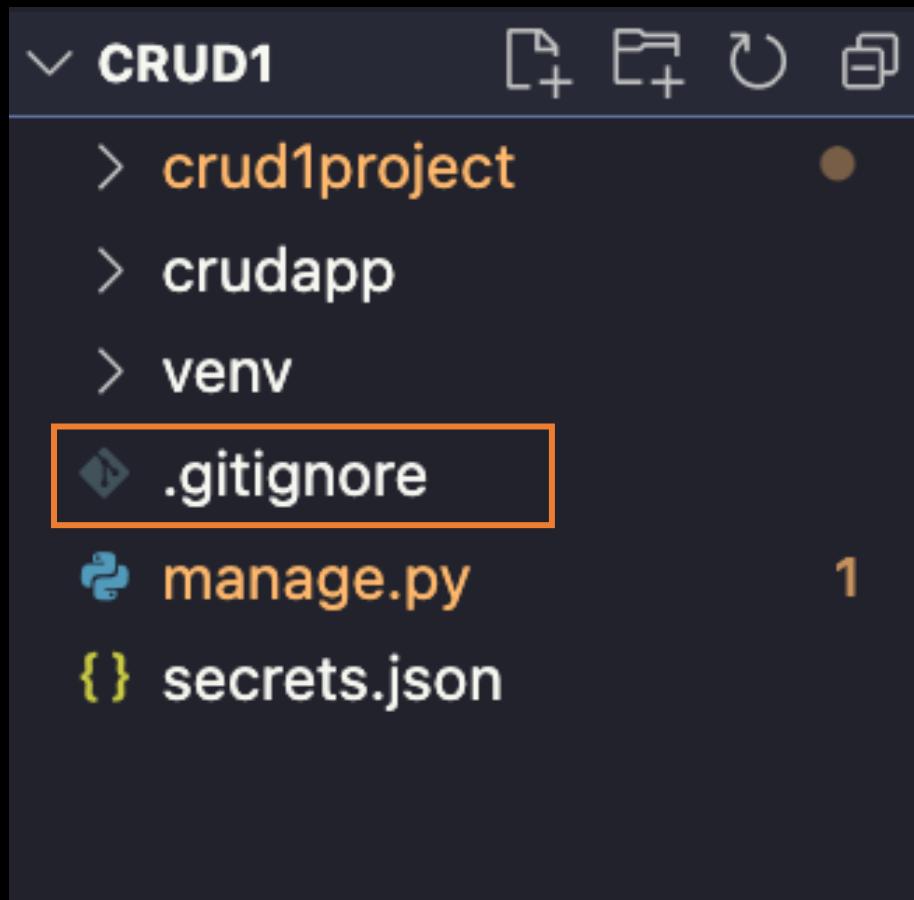
# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST
```

**Ctrl + A 해서 전체 선택
Ctrl + C 하여 복사해주세요**

〈환경설정〉

13. manage.py 와 같은 위치에 .gitignore 파일을 만들고
복사한 내용을 붙여넣기 한 후 secrets.json 을 추가로 작성해준 후 저장해주세요.



```
◆ .gitignore
1  # Created by https://www.toptal.com/developers/gitignore/api/python,django
2  # Edit at https://www.toptal.com/developers/gitignore?templates=python,django
3
4  ### Django ####
5  *.log
6  *.pot
7  *.pyc
8  __pycache__/
9  local_settings.py
10 db.sqlite3
11 db.sqlite3-journal
12 media
13 secrets.json
14
```

secrets.json 을 꼭 !!!
작성해주세요.

CR
not form

CRUD

〈 CR – not form 〉

Django 에서 사용자에게 데이터를 입력받는 세 가지 방법이 있는데요

- HTML Form 이용하는 방식
- Django Form 이용하는 방식
- Django modelForm 이용하는 방식

우선 첫번째 실습은 django의 form 을 사용하지 않고
html 코드를 통해 입력을 받아보도록 할게요.

〈 CR _ not form 〉

crudapp > models.py > Blog > __str__

crudapp / models.py

```
1  from django.db import models  
2  
3  # Create your models here.  
4  
5  class Blog(models.Model):  
6      title = models.CharField(max_length=100)  
7      pub_date = models.DateTimeField('data published')  
8      body = models.TextField()  
9  
10     def __str__(self):  
11         return self.title
```

이렇게 작성한 후 저장해주세요.
저장 필수 !!!

〈 CR _ not form 〉

```
5  class Blog(models.Model):                      crudapp / models.py
6      title = models.CharField(max_length=100)
7      pub_date = models.DateTimeField('data published')
8      body = models.TextField()
```

Django에서 Field는
테이블에서 열에 저장할 데이터의 유형을 제시해주고
이따 배울 품에 적용될 최소요구사항을 제시해주는 역할을 해요.

즉, 우리가 입력하는 데이터와 아주 밀접한 관계를 가지는 것으로
데이터의 종류에 맞춰 Field를 작성해주어야 해요 !

〈 CR – not form 〉

Windows :: python manage.py makemigrations

Mac :: python3 manage.py makemigrations

를 작성해주세요

```
● (venv) youngbeen@YoungBeenui-MacBookAir ➜ ~/Desktop/crud1 ➜ python3 manage.py makemigrations
Migrations for 'crudapp':
crudapp/migrations/0001_initial.py
  - Create model Blog
```

이는 **python**으로 작성한 코드를 DB가 알아들을 수 있도록
models.py에서 우리가 작성한 코드를 번역해주는 명령어예요.

〈 CR – not form 〉

Windows :: python manage.py migrate

Mac :: python3 manage.py migrate

를 작성해주세요

```
● (venv) youngbeen@YoungBeenui-MacBookAir ~ ~/Desktop/crud1▶ python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, crudapp, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying crudapp.0001_initial... OK
  Applying sessions.0001_initial... OK
```

이는 **python**으로 작성한 코드를
DB가 알아들을 수 있도록

models.py에 작성된 내용을
DB에 적용하는 명령어입니다.

〈 CR – not form 〉

Windows :: python manage.py createsuperuser

Mac :: python3 manage.py createsuperuser

를 작성해주세요

- (venv) youngbeen@YoungBeenui-MacBookAir ~ ~/Desktop/crud1 ➔ python3 manage.py createsuperuser
Username (leave blank to use 'youngbeen'): admin
Email address:
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y **Username : admin**
Superuser created successfully.

Email address : 그냥 엔터

Password / Password (again) : 1234
마지막은 y 치고 엔터

〈CR – not form〉

crudapp >  admin.py

```
1 from django.contrib import admin  
2 from .models import Blog  
3 # Register your models here.  
4  
5 admin.site.register(Blog)
```

crudapp 의 admin.py 에 이렇게 코드를 작성하고 저장해주세요.
이를 통해 admin 페이지에 Blog 클래스를 등록할 수 있습니다.

{ CR – not form }

✓ CRUD1

```
> crud1project      •  
  ✓ crudapp          •  
    > __pycache__  
    > migrations  
  ✓ templates  
    <> index.html  
  ✘ __init__.py  
  ✘ admin.py         2  
  ✘ apps.py  
  ✘ models.py        2  
  ✘ tests.py  
  ✘ views.py  
> venv  
❖ .gitignore  
≡ db.sqlite3  
✿ manage.py  
{ secrets.json
```

crudapp > templates > <> index.html > html

```
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4    <meta charset="UTF-8">  
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7    {% block title %}  
8      <title>Index</title>  
9    {% endblock %}  
10   </head>  
11   <body>  
12     {% block content %}  
13       <h1>Index Page</h1>  
14       <hr>  
15     {% endblock %}  
16   </body>  
17 </html>
```

crudapp / templates 폴더에 index.html 을 만들고
이렇게 작성해주세요.

〈CR – not form〉

crudapp > 🐍 views.py > 🏷 index

```
1  from django.shortcuts import render
2
3  # Create your views here.
4  def index(request):
5      return render(request, 'index.html')
```

crudapp / views.py 에
view 로직을 작성해줄게요.

〈CR – not form〉

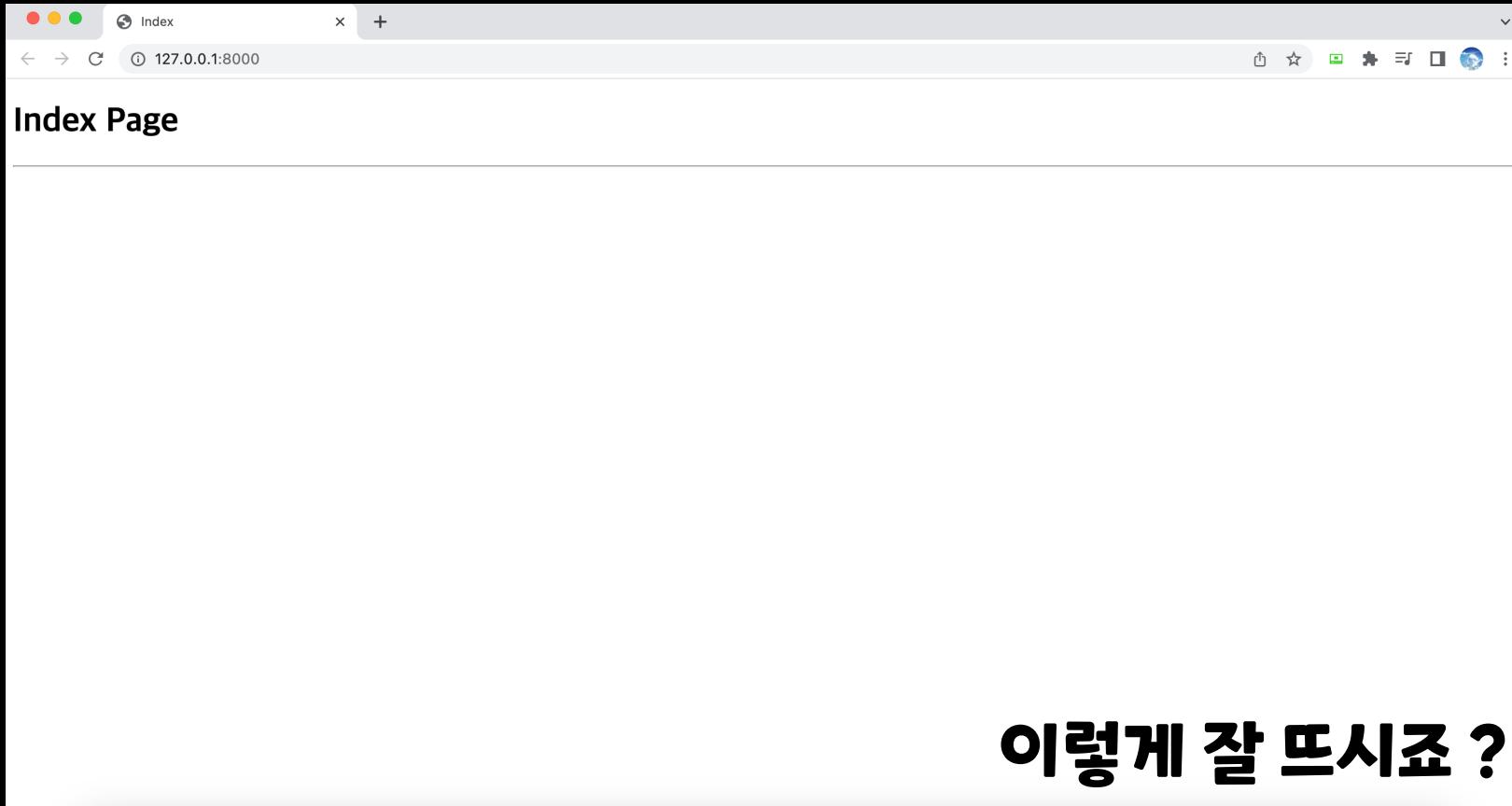
crud1project >urls.py > ...

```
16  from django.contrib import admin  
17  from django.urls import path  
18  import crudapp.views  
19  
20 urlpatterns = [  
21     path('admin/', admin.site.urls),  
22     path('', crudapp.views.index, name = 'index'),  
23 ]  
24
```

crud1project / urls.py 에
url 연결을 위한 코드를 작성해줄게요.

〈 CR – not form 〉

**Windows :: python manage.py runserver
Mac :: python3 manage.py runserver
로 런서버 해줄게요**



〈 CR – not form 〉

① 127.0.0.1:8000/admin
② 127.0.0.1:8000/admin
③ 127.0.0.1:8000/admin - Google 검색

주소창에 <http://127.0.0.1:8000>

뒤에 /admin 을 작성해서 admin 페이지로 이동

Django administration

Username:
admin **Username : admin**

Password:
.... **Password : 1234**

Log in

로그인 해주세요

← → ⌂ ⓘ 127.0.0.1:8000/admin/

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	Change
Users	+ Add	Change

CRUDAPP

Blogs	+ Add	Change
-------	-------	--------

이렇게 되면 성공 !

{ CR – not form }

Windows :: ctrl + c

Mac :: control + c

로 서버를 종료해주세요

```
● (venv) youngbeen@YoungBeenui-MacBookAir ➜ ~/Desktop/crud1 ➔ python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

System check identified no issues (0 silenced).

May 17, 2023 - 05:08:47

Django version 4.2.1, using settings 'crud1project.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CONTROL-C.

[17/May/2023 05:08:49] "GET / HTTP/1.1" 200 320

Not Found: /favicon.ico

[17/May/2023 05:08:49] "GET /favicon.ico HTTP/1.1" 404 2230

^C

```
○ (venv) youngbeen@YoungBeenui-MacBookAir ➜ ~/Desktop/crud1 ➔
```

〈 CR _ not form 〉

Form을 사용하지 않으면
Create를 구현하기 위해 두 가지 기능이 필요해요

NEW

NEW는 글 쓰는 페이지를 보여주는 기능 을 해요

NEW가 html 에서 글 작성 form을 작성해준다고 생각해주면 되어요!

CREATE

CREATE는 NEW를 통해 작성한 글의
데이터베이스를 저장하는 역할 을 합니다.

즉 NEW 통해 데이터를 사용자에게서 입력받고,
CREATE로 저장 하는 것이 되겠죠?

{ CR _ not form }

The image shows a code editor interface with a sidebar containing project files and a main pane displaying the content of the 'new.html' file. The 'new.html' file is highlighted with an orange border.

```
1  {% extends 'index.html' %}\n2\n3  {% block title %}\n4      <title>New</title>\n5  {% endblock %}\n6\n7  {% block content %}\n8      <h1>New Page</h1>\n9      <hr>\n10     <form method="POST" action="{% url 'create' %}">\n11         {% csrf_token %}\n12         제목 : <input type="text" name="title"> <br>\n13         본문 : <textarea name="body" cols="30" rows="10"></textarea> <br>\n14         <button type = "submit"> 작성</button>\n15     </form>\n16  {% endblock %}
```

crudapp / templates / new.html 을 생성하고
이렇게 작성해주세요.

{ CR – not form }

crudapp > templates > new.html > form > br

```
1  {% extends 'index.html' %}  
2  
3  {% block title %}  
4      <title>New</title>  
5  {% endblock %}  
6  
7  {% block content %}  
8      <h1>New Page</h1>  
9      <hr>  
10     <form method="POST" action="{% url 'create' %}">  
11         {% csrf_token %}  
12         제목 : <input type="text" name="title"> <br>  
13         본문 : <textarea name="body" cols="30" rows="10"></textarea> <br>  
14         <button type = "submit"> 작성</button>  
15     </form>  
16  {% endblock %}
```

form 의 method 를 POST 로,
action 은 create 와 연결해주었어요.

〈 CR – not form 〉

HTTP Method

: form을 처리할 때는
같은 URL에서 GET 방식 / POST 방식으로 나눠서 처리하는데요

GET :: 디폴트 값으로, 데이터 변경 없이 단순한 요청
데이터가 URL에 보이기 때문에 보안에 취약합니다.

POST :: 입력된 데이터를 저장하여 서버로 전송
데이터가 URL에 보이지 않습니다.

Post로 폼 데이터 보낼 때는 { % CSRF_TOKEN % } 함께 작성해야합니다.

〈 CR – not form 〉

{% CSRF_TOKEN %}

CSRF 공격에 대비하기 위해 작성하는 코드로,
공격자가 의도한 행위를 특정 웹사이트가 요청하게 하는 공격
이것을 방지하기 위해

장고에서는 {% CSRF_TOKEN %} 을 작성하여
토큰을 검사함으로써 대비해요

:: 즉, 웹 사이트의 보안을 위해 작성한다 !

{ CR – not form }

crudapp > templates > new.html > form > br

crudapp / templates / new.html

```
1  {% extends 'index.html' %}  
2  
3  {% block title %}      우리가 작성해준 input 태그와 textarea 태그입니다.  
4      <title>New</title>  
5  {% endblock %}          유저가 글을 작성하는 폼을  
6  
7  {% block content %}    제작해주기 위해서 작성한 코드에요.  
8      <h1>New Page</h1>  
9      <hr>  
10     <form method="POST" action="{% url 'create' %}">  
11         {% csrf_token %}          여기서 name을 지정해주어야  
12         제목 : <input type="text" name="title"> <br>  views.py에서 데이터를 가져올 수 있어요 !  
13         본문 : <textarea name="body" cols="30" rows="10"></textarea> <br>  
14         <button type = "submit"> 작성</button>  
15     </form>  
16  {% endblock %}
```

우리가 작성해준 input 태그와 textarea 태그입니다.
유저가 글을 작성하는 폼을
제작해주기 위해서 작성한 코드에요.
여기서 name을 지정해주어야
views.py에서 데이터를 가져올 수 있어요 !

{ CR – not form }

crudapp > templates > new.html

crudapp / templates / new.html

```
1  {% extends 'index.html' %}  
2  
3  {% block title %}  
4      <title>New</title>  
5  {% endblock %}  
6  
7  {% block content %}  
8      <h1>New Page</h1>  
9      <hr>  
10     <form method="POST" action="{% url 'create' %}">  
11         {% csrf_token %}  
12         제목 : <input type="text" name="title"> <br>  
13         본문 : <textarea name="body" cols="30" rows="10"></textarea> <br>  
14         <button type = "submit"> 작성</button>  
15     </form>  
16  {% endblock %}
```

작성한 데이터를 저장해주기 위한
Submit 태입의 button 코드입니다

{ CR – not form }

crudapp > 📄 views.py > ⚒ create

crudapp / views.py

```
1  from django.shortcuts import render, redirect
2  from django.utils import timezone
3  from .models import Blog
4
5  # Create your views here.
6  def index(request):
7      return render(request, 'index.html')
8
9  def new(request):
10     return render(request, 'new.html')
11
12 def create(request):
13     blog = Blog()
14     blog.title = request.POST['title']
15     blog.pub_date = timezone.now()
16     blog.body = request.POST['body']
17     blog.save()
18     return redirect('read')
```

〈CR – not form〉

crud1project > urls.py > ...

crud1project/ urls.py

```
16  from django.contrib import admin
17  from django.urls import path
18  import crudapp.views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', crudapp.views.index, name = 'index'),
23     path('new/', crudapp.views.new, name = 'new'),
24     path('new/create/', crudapp.views.create, name = 'create'),
25 ]
26 ]
```

이렇게 작성해주세요.

〈 CR – not form 〉

crudapp > views.py > create

```
1  from django.shortcuts import render, redirect
2  from django.utils import timezone
3  from .models import Blog
4
5  # Create your views here.
6  def index(request):
7      return render(request, 'index.html')
8
9  def new(request):
10     return render(request, 'new.html')
11
12 def create(request):
13     blog = Blog()
14     blog.title = request.POST['title']
15     blog.pub_date = timezone.now()
16     blog.body = request.POST['body']
17     blog.save()
18     return redirect('read')
```

crudapp / views.py

저희가 create 는 new 와는 달리
따로 template 을 생성해주지 않았어요.

create 함수는
new를 통해 전달받은 데이터를 추가하고
저장해주는 함수를 작성해준 것입니다.

더 자세한 코드리뷰를 해볼게요

〈 CR – not form 〉

```
crudapp > 🗂 views.py > ⚒ create  
1  from django.shortcuts import render, redirect  
2  from django.utils import timezone  
3  from .models import Blog  
4  
5  # Create your views here.  
6  def index(request):  
7      return render(request, 'index.html')  
8  
9  def new(request):  
10     return render(request, 'new.html')  
11  
12 def create(request):  
13     blog = Blog()  
14     blog.title = request.POST['title']  
15     blog.pub_date = timezone.now()  
16     blog.body = request.POST['body']  
17     blog.save()  
18     return redirect('read')
```

crudapp / views.py

Blog 클래스의 객체를 생성하고
POST 요청 방식으로 전달된
데이터에서 값을 추출해서
객체의 속성을 설정 줍니다.

또한 import 해온 timezone의
now 메소드를 통해 현재 시간을 받아와
모델 테이블에 저장합니다 ~

save() 메소드를 사용해주어야
객체 정보가 모델 테이블에 저장이 되겠죠?

{ CR – not form }

crudapp > 📄 views.py > ⚒ create

```
1  from django.shortcuts import render, redirect
2  from django.utils import timezone
3  from .models import Blog
4
5  # Create your views here.
6  def index(request):
7      return render(request, 'index.html')
8
9  def new(request):
10     return render(request, 'new.html')
11
12 def create(request):
13     blog = Blog()
14     blog.title = request.POST['title']
15     blog.pub_date = timezone.now()
16     blog.body = request.POST['body']
17     blog.save()
18     return redirect(['read'])
```

crudapp / views.py

여기서 **redirect**는
새로운 페이지가 아닌
기존 페이지간의 이동에
사용됩니다.

{ CR – not form }

crud1project > urls.py > ...

crud1project/ urls.py

```
16  from django.contrib import admin
17  from django.urls import path
18  import crudapp.views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', crudapp.views.index, name = 'index'),
23     path('new/', crudapp.views.new, name = 'new'),
24     path('new/create/', crudapp.views.create, name = 'create'),
25 ]
26 
```

이 코드를 잘 보면

{ CR – not form }

crudapp > templates > new.html > form > br

```
1  {% extends 'index.html' %}  
2  
3  {% block title %}  
4      <title>New</title>  
5  {% endblock %}  
6  
7  {% block content %}  
8      <h1>New Page</h1>  
9      <hr>  
10     <form method="POST" action="{% url 'create' %}">  
11         {% csrf_token %}  
12         제목 : <input type="text" name="title"> <br>  
13         본문 : <textarea name="body" cols="30" rows="10"></textarea> <br>  
14         <button type = "submit"> 작성</button>  
15     </form>  
16  {% endblock %}
```

crudapp / templates / new.html에서
action 속성을 통해
요청이 create로 전달되도록
지정해준 것이 되겠네요 !

CRUD

{ CR – not form }

crudapp / templates / read.html

CRUD1

> crud1project

crudapp

> __pycache__

> migrations

templates

<> index.html

<> new.html

<> read.html

__init__.py

admin.py

apps.py

models.py

tests.py

views.py

crudapp > templates > <> read.html > ...

```
1  {% extends 'index.html' %}\n2\n3  {% block title %}\n4      <title>Read</title>\n5  {% endblock %}\n6\n7  {% block content %}\n8      <h1>Read Page</h1>\n9      <hr>\n10     {% for blogs in blogs.all %}\n11         <h3>{{ blogs.title }}</h3>\n12         <p>{{ blogs.pub_date }}</p>\n13         <p>{{ blogs.body }}</p>\n14     {% endfor %}\n15     {% endblock %}
```

crudapp / templates 폴더에
read.html 을 만들고 코드를 작성해주세요

{ CR - not form }

```
10     return redirect('read')
11
12
13
14
15
16
17
18
19
20 def read(request):
21     blogs = Blog.objects
22     return render(request, 'read.html', {'blogs': blogs})
```

crudapp / views.py

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23 path('new/', crudapp.views.new, name = 'new'),
24 path('new/create/', crudapp.views.create, name = 'create'),
25 path('read/', crudapp.views.read, name = 'read'),
26 ]
27
```

crud1project / urls.py

각각 잘 작성해주세요

{ CR _ not form }

crudapp > templates > index.html > html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1
7      {% block title %}
8          <title>Index</title>
9      {% endblock %}
10     </head>
11     <body>
12         <nav>
13             <a href="{% url 'new' %}">글 작성하기</a>
14             <a href="{% url 'read' %}">글 모아보기</a>
15         </nav>
16         {% block content %}
17             <h1>Index Page</h1>
18             <hr>
19         {% endblock %}
20     </body>
21     </html>
```

crudapp / templates / index.html

페이지간 수월한 이동을 위해
nav 바를 작성해줄게요

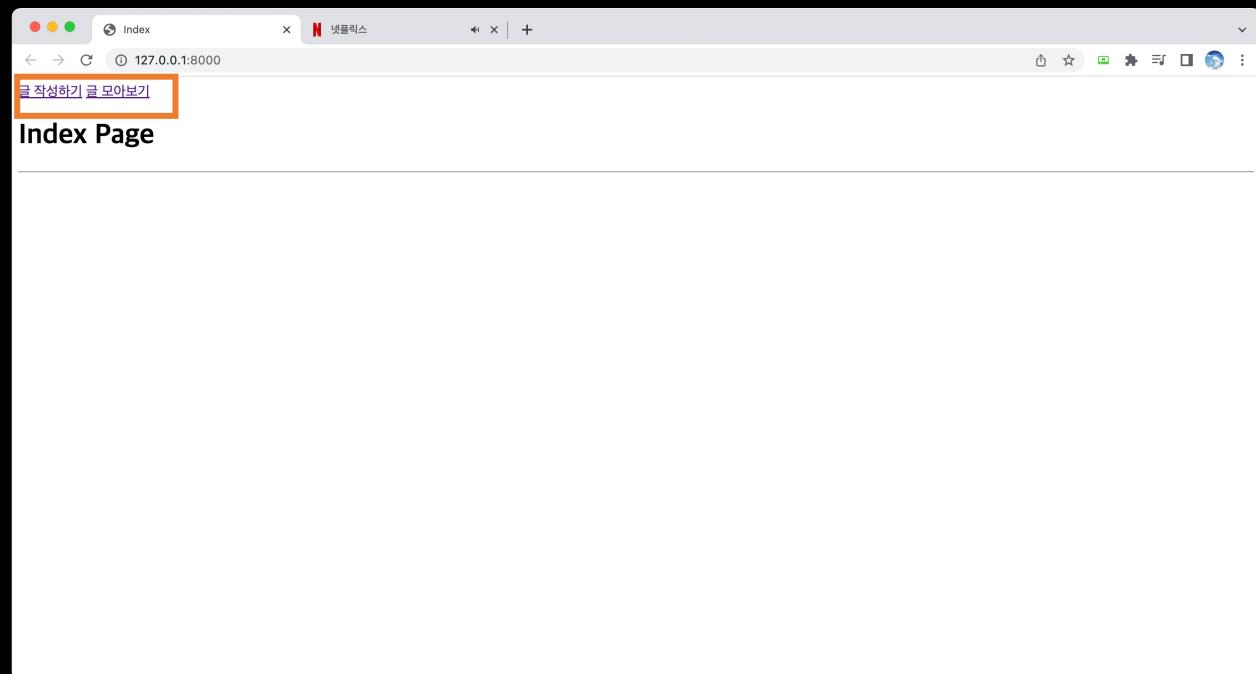
〈 CR – not form 〉

Windows :: python manage.py runserver

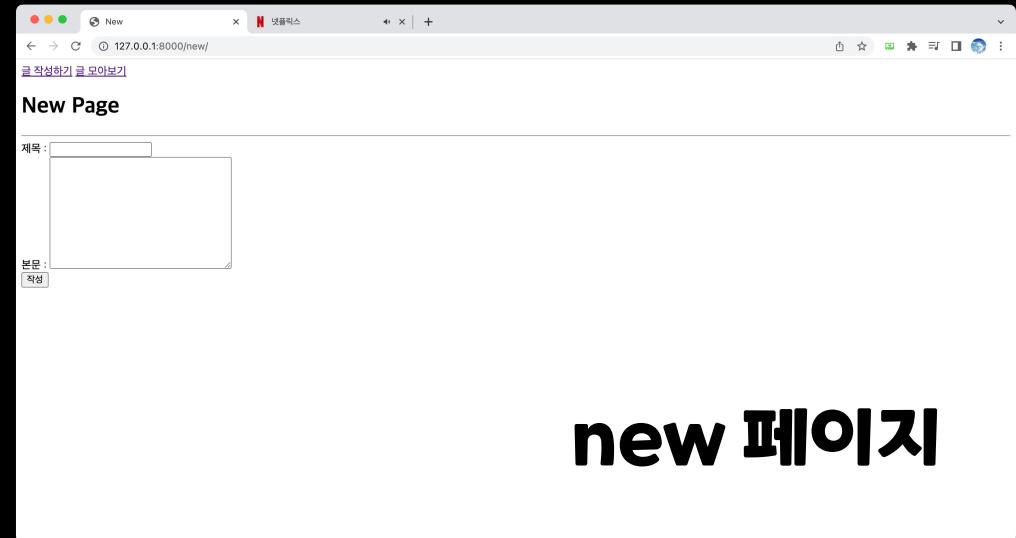
Mac :: python3 manage.py runserver

런서버 해줄게요

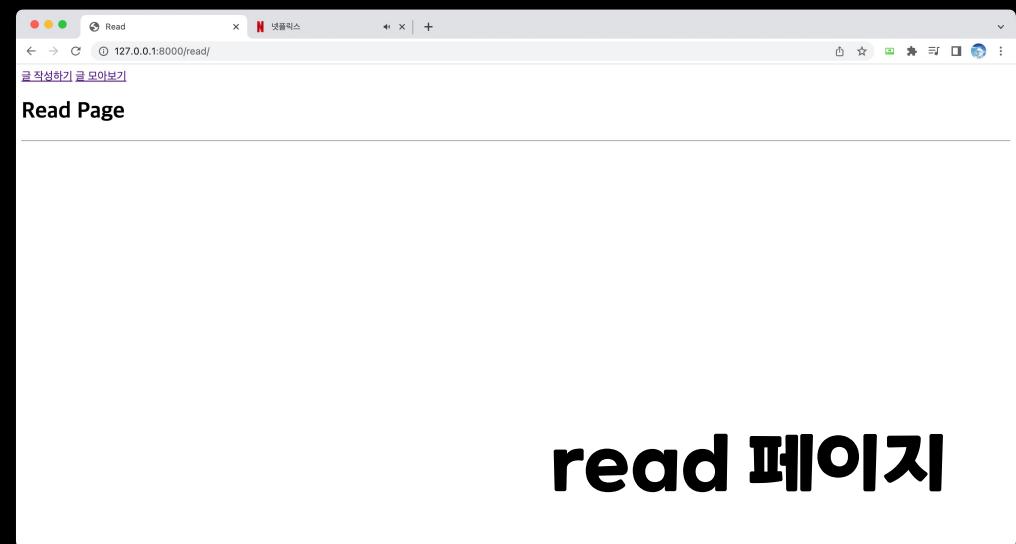
⟨ CR – not form ⟩



**런서버 하면 이렇게 뜨시나용?
그럼 위에 nav바를 눌러서
오른쪽과 같이 오류없이
이동이 잘 되는지 봐주세요**



new 페이지



read 페이지

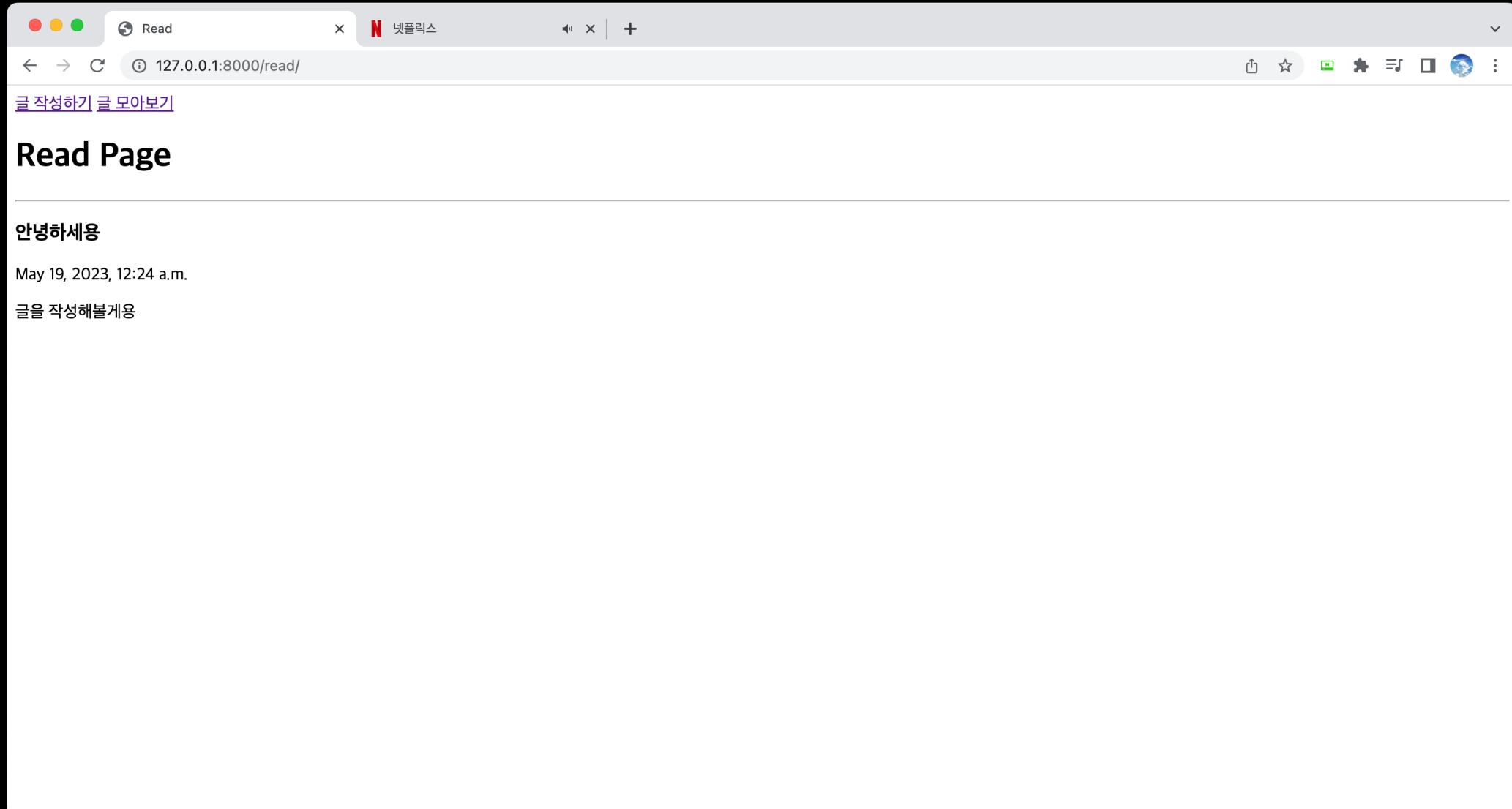
〈 CR – not form 〉

A screenshot of a web browser window titled "New". The address bar shows "127.0.0.1:8000/new/". The page content is titled "New Page". It contains a form with the following fields:

- 제목 :
- 글을 작성해볼게용
[Large text area]
- 본문 :

이렇게 작성해주고 작성 버튼 눌러주세요

⟨ CR – not form ⟩



그럼 이렇게 잘 작성되죠!?

〈 CR – not form 〉

The screenshot shows the Django admin interface at <http://127.0.0.1:8000/admin/>. The 'CRUDAPP' section is highlighted with an orange border. The 'Blogs' section is also highlighted with an orange border.

<http://127.0.0.1:8000/admin/>

로 이동해서
아까 등록한 부분을 눌러줄게요

The screenshot shows the 'Select blog to change' page in the Django admin site. The 'BLOG' section is expanded, showing a single entry: '안녕하세요'. This entry is highlighted with an orange border.

The screenshot shows the 'Change blog' page for the blog entry '안녕하세요'. The 'Title' field contains '안녕하세요'. The 'Data published:' section shows 'Date: 2023-05-19 Today' and 'Time: 00:24:53 Now'. The 'Body' field contains '글을 작성해볼게용'. At the bottom, there are buttons for 'SAVE', 'Save and add another', 'Save and continue editing', and a red 'Delete' button.

잘 뜨시죠?

⟨ CR – not form ⟩

Windows :: ctrl + c

Mac :: control + c

서버 종료 해주세요

{ CR – not form }

crudapp / templates / detail.html

CRUD1

crud1project

__pycache__

__init__.py

asgi.py

settings.py

urls.py

wsgi.py

crudapp

__pycache__

migrations

templates

detail.html

crudapp > templates > detail.html > ...

```
1  {% extends 'index.html' %}\n2\n3  {% block title %}\n4      <title>Detail</title>\n5  {% endblock %}\n6\n7  {% block content %}\n8      <h1>Detail Page</h1>\n9      <hr>\n10     <h3>{{ blog.title }}</h3>\n11     <p>{{ blog.pub_date }}</p>\n12     <p>{{ blog.body }}</p>\n13  {% endblock %}
```

crudapp / template 폴더에
detail.html을 작성해주세요

〈 CR – not form 〉

crudapp >  views.py >  detail

```
1  from django.shortcuts import render, redirect, get_object_or_404
2  from django.utils import timezone
3  from .models import Blog
```

23
24
25
26

```
def detail(request, id):
    blog = get_object_or_404(Blog, id = id)
    return render(request, 'detail.html', {'blog': blog})
```

crudapp / views.py 에 작성해주세요.

get_object_or_404 는

Blog모델에서 id값을 blog에 담아 가져올텐데
가져오지 못하면 404에러를 일으키는 기능을 해요

⟨ CR – not form ⟩

crud1project > urls.py > ...

```
20     urlpatterns = [
21         path('admin/', admin.site.urls),
22         path('', crudapp.views.index, name = 'index'),
23         path('new/', crudapp.views.new, name = 'new'),
24         path('new/create/', crudapp.views.create, name = 'create'),
25         path('read/', crudapp.views.read, name = 'read'),
26         path('detail/<str:id>/', crudapp.views.detail, name = 'detail'),
27     ]
```

crud1project / urls.py 에 detail url 을 위한 코드를 작성해줄게요.
이는 특정 게시물을 가져와야 하기 때문에
url에도 id 값이 필요해요

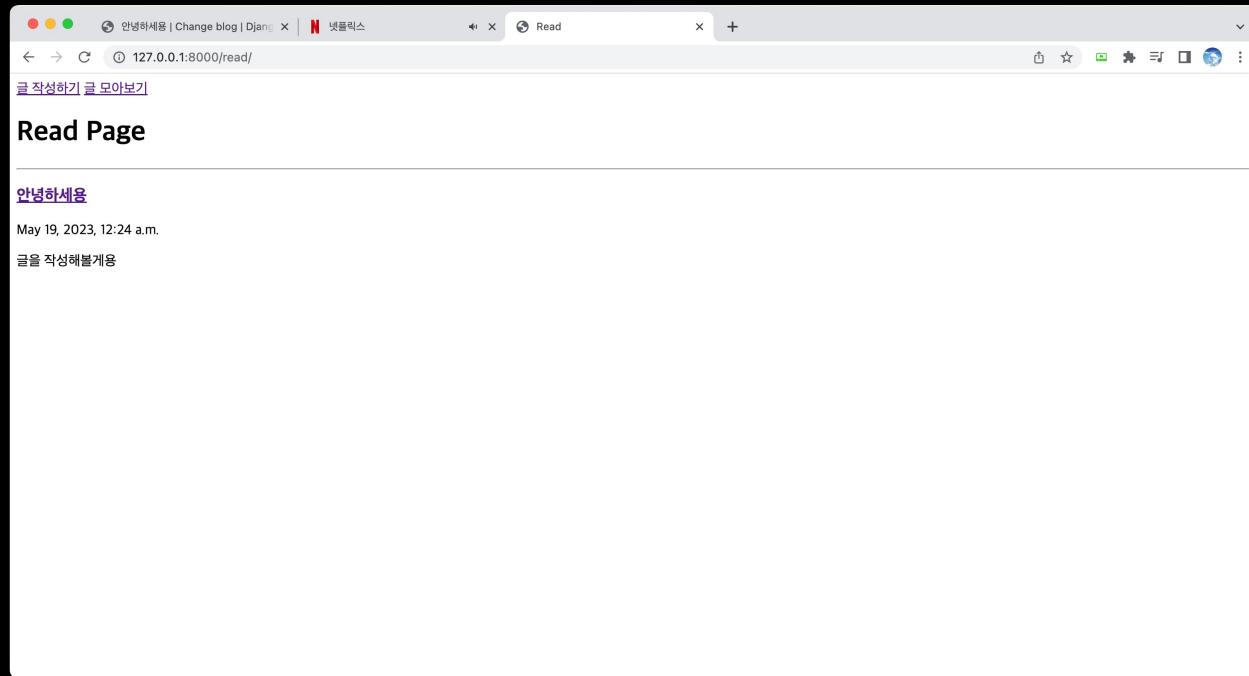
〈 CR – not form 〉

Windows :: python manage.py runserver

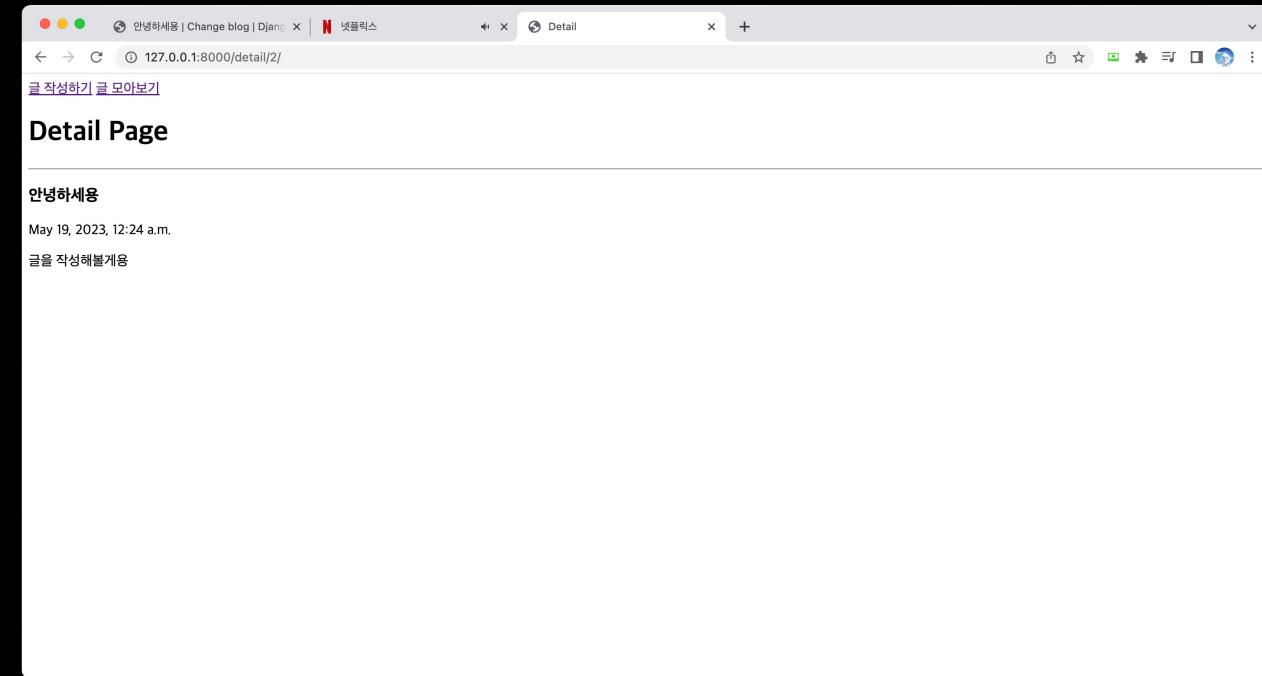
Mac :: python3 manage.py runserver

다시 런서버 해줄게요

〈 CR – not form 〉



**Read 페이지로 가면 html 코드에 추가해준
Detail 페이지로 가는 a 코드가 추가되어 있어요
클릭해줄게요**



**그러면 이렇게 Detail 페이지로 이동하며
선택한 글의 내용이 잘 떠요 !**

⟨ CR – not form ⟩

Windows :: ctrl + c

Mac :: control + c

서버 종료 해주세요

CR
use form

〈 CR – use form 〉

이번에는 django 에서 제공해주는 modelform 을 사용하여
CRUD 기능을 구현해볼거에요.

Django modelform 은
forms.py 파일을 앱 폴더에 작성해주어야해요

Django form 을 사용하면
아까 html form 을 사용하는 것에 비해 여러 장점이 있어요.

〈 CR – use form 〉

form을 사용하면
Django에서 입력폼도 html에 자동으로 생성해주고
입력폼의 값을 알아서 검증하고 변환해주거든요 !

이 때 검증을 통과한 값들은 딕셔너리 타입으로 제공됩니다.

그럼 이런 편리한 걸 왜 먼저 안 배웠냐면요 ..

〈 CR – use form 〉

저희가 Django만을 가지고 웹 개발을 한다면
매우매우 효율적이고 좋겠지만

저희는 추후 해커톤에서 저희가 작성한 품에
css도 먹이고 javascript도 사용해야해요.

자동으로 작성해주는 입력폼이 편리하긴 하지만,
내 마음대로 커스터마이징을 하기가 어렵다는 단점이 있어요 !!
따라서 이러한 이유 때문에
지금 알려드릴 form을 사용하는 방식과
form을 사용하지 않는 방식 둘 다 알아두셔야
마음껏 골라서 사용할 수 있겠죠 ??

〈환경설정〉

1. 바탕화면에 crud2 폴더를 생성 후 Vscode에서 열어주세요

2. Vscode 터미널을 열어
가상환경을 생성해주세요

Windows :: **python -m venv venv**

Mac :: **python3 -m venv venv**

3. 가상환경을 실행해주세요

Windows :: **source venv/Scripts/activate**

Mac :: **source venv/bin/activate**

4. Django를 설치해주세요.

Windows & Mac :: **pip install django**

〈환경설정〉

5. ‘crud2project’ 프로젝트를 생성해주세요.

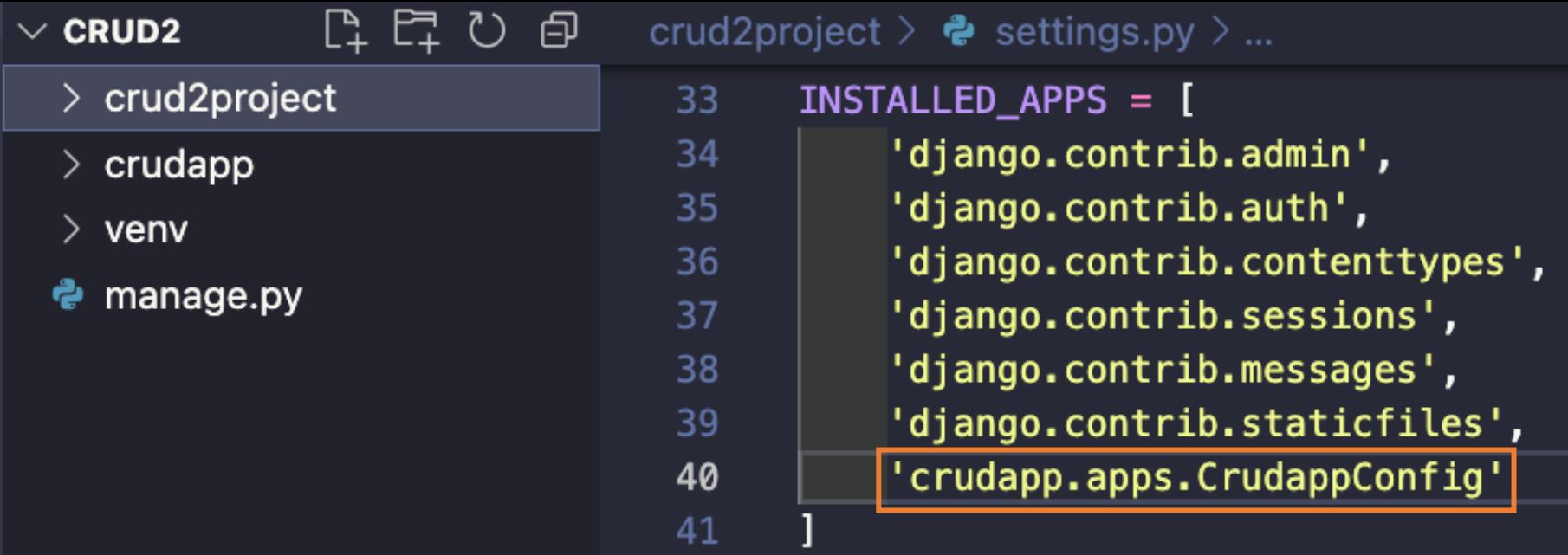
Windows & Mac :: django-admin startproject crud2project .

6. ‘crudapp’ 앱을 생성해주세요

Windows :: python manage.py startapp crudapp

Mac :: python3 manage.py startapp crudapp

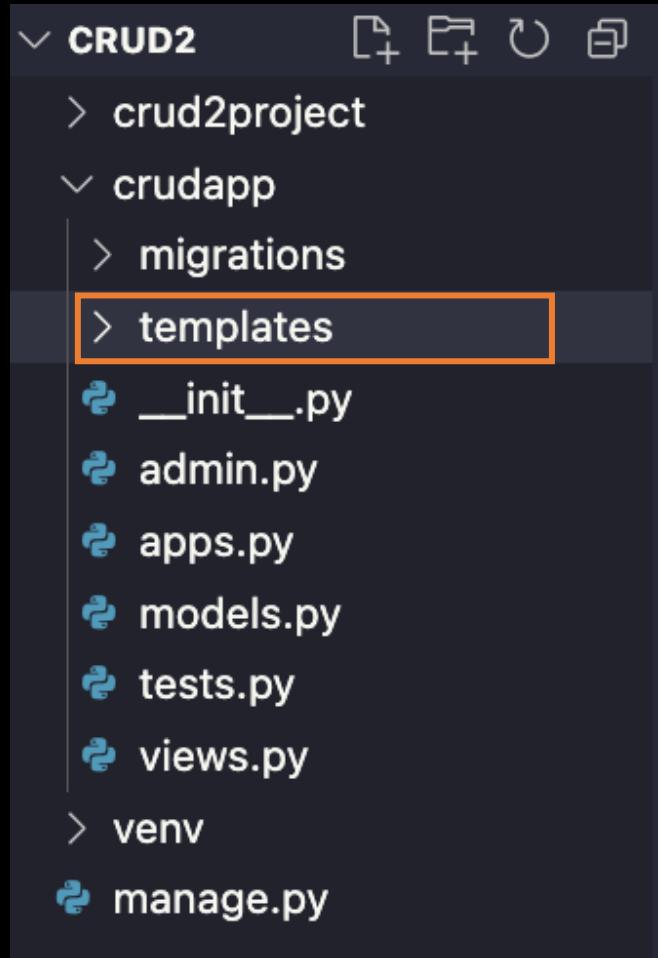
7. ‘crud2project’ 의 settings.py 에 생성한 ‘crudapp’ 정의하기



```
crud2project > settings.py > ...
33     INSTALLED_APPS = [
34         'django.contrib.admin',
35         'django.contrib.auth',
36         'django.contrib.contenttypes',
37         'django.contrib.sessions',
38         'django.contrib.messages',
39         'django.contrib.staticfiles',
40         'crudapp.apps.CrudappConfig'
41     ]
```

〈환경설정〉

8. 'crudapp' 속에 templates 폴더 생성



9. 'crud2project' 프로젝트 폴더의 settings.py에서 secret key를 복사해서

```
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = 'django-insecure-+7kdl%sby7zc_aryw&^34*n@^#-0g30q*hl8hg+f7t9#=+q7k'
24
```

manage.py 와 같은 위치에
secrets.json 파일을 만들고
이렇게 작성해주세요.

```
{}
1  {
2      "SECRET_KEY" : "django-insecure-+7kdl%sby7zc_aryw&^34*n@^#-0g30q*hl8hg+f7t9#=+q7k"
3 }
```

{
"SECRET_KEY" : "settings.py의 secret key 값"
}

〈환경설정〉

10. 'crud2project' 프로젝트 폴더의 settings.py에 import 해주고

crud2project > 🐍 settings.py > ...

```
13  from pathlib import Path
14  import os, json
15  from django.core.exceptions import ImproperlyConfigured
16  |
```

〈환경설정〉

11. secrets.json에서 secret key를 불러오기 위한 코드를 작성해줄게요.

crud2project > 🐍 settings.py > ...

```
24 # SECURITY WARNING: keep the secret key used in production secret!
25 secret_file = os.path.join(BASE_DIR, 'secrets.json')
26
27 with open(secret_file) as f:
28     secrets = json.loads(f.read())
29
30 def get_secret(setting, secrets = secrets):
31     try:
32         return secrets[setting]
33     except KeyError:
34         error_msg = "Set the {} enviroment variable".format(setting)
35         raise ImproperlyConfigured(error_msg)
36
37 SECRET_KEY = get_secret("SECRET_KEY")
38
```

〈환경설정〉

12. Google에 gitignore.io를 검색하고 Python, Django를 선택하여 생성한 문서를 복사한 후

› **gitignore.io**

자신의 프로젝트에 꼭 맞는 .gitignore 파일을 만드세요



```
# Created by https://www.toptal.com/developers/gitignore/api/python,django
# Edit at https://www.toptal.com/developers/gitignore?templates=python,django

### Django ####
*.log
*.pot
*.pyc
__pycache__/
local_settings.py
db.sqlite3
db.sqlite3-journal
media

# If your build process includes running collectstatic, then you probably don't need or want to include staticfiles/
# in your Git repository. Update and uncomment the following line accordingly.
# <django-project-name>/staticfiles/

### Django-Python Stack ####
# Byte-compiled / optimized / DLL files
*.py[co]d
*py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST
```

**Ctrl + A 해서 전체 선택
Ctrl + C 하여 복사해주세요**

〈환경설정〉

13. manage.py 와 같은 위치에 .gitignore 파일을 만들고
복사한 내용을 붙여넣기 한 후 secrets.json 을 추가로 작성해준 후 저장해주세요.

```
▽ CRUD2
  > crud2project
  > crudapp
  > venv
  ◇ .gitignore
  ✎ manage.py
  {} secrets.json

    ◇ .gitignore
      4   ### Django ####
      5   *.log
      6   *.pot
      7   *.pyc
      8   __pycache__/
      9   local_settings.py
     10  db.sqlite3
     11  db.sqlite3-journal
     12  media
     13  secrets.json
      14
      15  # If your build process includes running collectstatic, the
      16  # in your Git repository. Update and uncomment the following
      17  # <django-project-name>/staticfiles/
      18
```

**secrets.json 을 꼭 !!!
작성해주세요.**

CRUD

〈 CR _ use form 〉

crudapp > models.py > Blog > __str__

crudapp / models.py

```
1  from django.db import models  
2  
3  # Create your models here.  
4  
5  class Blog(models.Model):  
6      title = models.CharField(max_length=100)  
7      pub_date = models.DateTimeField('data published')  
8      body = models.TextField()  
9  
10     def __str__(self):          아까와 동일하게 작성해줄게요  
11         return self.title
```

〈 CR – use form 〉

Windows :: python manage.py makemigrations
Mac :: python3 manage.py makemigrations
작성한 후

Windows :: python manage.py migrate
Mac :: python3 manage.py migrate
를 작성해주세요

〈 CR – use form 〉

Windows :: **python manage.py createsuperuser**

Mac :: **python3 manage.py createsuperuser**
를 작성해주세요

- youngbeen@YoungBeenui-MacBookAir ✨ ~/Desktop/crud2 ➔ python3 manage.py createsuperuser
Username (leave blank to use 'youngbeen'): admin
Email address:
Password: 1234
Password (again): 1234
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

Username : admin

Email : 엔터치기

Password : 1234

Password (again) : 1234

Bypass : y

〈 CR – use form 〉

crudapp / admin.py

CRUD2

> crud2project

✓ crudapp

> __pycache__

> migrations

> templates

🐍 __init__.py

🐍 admin.py

crudapp > 🐍 admin.py

```
1  from django.contrib import admin
2  from .models import Blog
3  # Register your models here.
4
5  admin.site.register(Blog)
```

동일하게 admin 페이지에 Blog 모델을
추가해줄게요

〈 CR – use form 〉

crudapp / forms.py

CRUD2

> crud2project

crudapp

> __pycache__

> migrations

> templates

forms.py

admin.py

apps.py

forms.py

crudapp > forms.py > BlogForm > Meta

```
1 from django import forms  
2 from .models import Blog  
3  
4 class BlogForm(forms.ModelForm):  
5     class Meta:  
6         model = Blog  
7         fields = ['title', 'body']
```

코드를 작성해줄게요

이렇게 crudapp 하위에 forms.py 를
만들어주고

〈 CR _ use form 〉

crudapp / forms.py

crudapp > forms.py > BlogForm > Meta

```
1  from django import forms  
2  from .models import Blog  
3  
4  class BlogForm(forms.ModelForm):  
5      class Meta:  
6          model = Blog  
7          fields = ['title', 'body']
```

우리가 작성해준 models.py의 Blog 모델을 사용하여
이를 django의 form 형식으로 사용해줄 것이기 때문에
이에 맞춰 Import 해주었어요

〈 CR _ use form 〉

crudapp / forms.py

crudapp > forms.py > BlogForm > Meta

```
1  from django import forms
2  from .models import Blog
3
4  class BlogForm(forms.ModelForm):
5      class Meta:
6          model = Blog
7          fields = ['title', 'body']
```

BlogForm 의 메타데이터를 위해 쓰인 코드예요.

메타데이터는 데이터에 대한 데이터로,

이 BlogForm의 model은 어떤 것이고 field는 어떤 것이고 등
BlogForm의 메타데이터를 위한 코드라고 이해하면 좋을 것 같아요.

〈 CR _ use form 〉

crudapp / forms.py

crudapp > forms.py > BlogForm > Meta

```
1  from django import forms
2  from .models import Blog
3
4  class BlogForm(forms.ModelForm):
5      class Meta:
6          model = Blog
7          fields = ['title', 'body']
```

BlogForm의 기반이 되는 model이 Blog 모델이며,
Blog 모델을 기반으로 폼을 생성할 것이다
라고 명시해주는 코드에요

〈CR_use form〉

crudapp / forms.py

crudapp > forms.py > BlogForm > Meta

```
1  from django import forms
2  from .models import Blog
3
4  class BlogForm(forms.ModelForm):
5      class Meta:
6          model = Blog
7          fields = ['title', 'body']
```

폼을 통해 어떤 필드를 입력받을지
작성해주는 코드입니다

{ CR _ use form }

crudapp / templates / index.html

CRUD2

```
> crud2project  
✓ crudapp  
> __pycache__  
> migrations  
✓ templates  
<> index.html  
~ __init__.py  
~ admin.py  
~ apps.py  
~ forms.py  
~ models.py  
~ tests.py  
~ views.py  
> venv
```

crudapp > templates > <> index.html > html > body

```
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4      <meta charset="UTF-8">  
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">  
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7      {% block title %}  
8          <title>Index</title>  
9      {% endblock %}  
10     </head>  
11     <body>  
12         {% block content %}  
13             <h1>Index Page</h1>  
14             <hr>  
15         {% endblock %}  
16     </body>  
17 </html>
```

crudapp / templates 폴더 안에
index.html을 만들고 코드를 작성해주세요

⟨ CR – use form ⟩

crudapp > 📁 views.py > 📁 index

crudapp / views.py

```
1  from django.shortcuts import render
2
3  # Create your views here.
4
5  def index(request):
6      return render(request, 'index.html')
```

crud2project > 📁 urls.py > ...

crud2project / urls.py

```
16  from django.contrib import admin
17  from django.urls import path
18  import crudapp.views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', crudapp.views.index, name = 'index'),
23 ]
```

{ CR _ use form }

crudapp / templates / create.html

CRUD2

crud2project

> __pycache__

__init__.py

asgi.py

settings.py

urls.py

wsgi.py

crudapp

> __pycache__

> migrations

templates

<> create.html

<> index.html

<> read.html

crudapp > templates > <> create.html > ...

```
1  {% extends 'index.html' %}\n\n2  {% block title %}\n3      <title>Create</title>\n4  {% endblock %}\n\n5  {% block content %}\n6      <h1>Create Page</h1>\n7      <hr>\n8      <form method="POST">\n9          {% csrf_token %}\n10         <table>\n11             {{ form.as_table }}\n12         </table>\n13         <button type = "submit">작성</button>\n14     </form>\n15\n16\n17  {% endblock %}
```

crudapp / templates 폴더에 create.html 을 만들고 작성해주세요

〈 CR – use form 〉

```
{% extends 'index.html' %}

{% block title %}
    <title>Create</title>
{% endblock %}

{% block content %}
    <h1>Create Page</h1>
    <hr>
    <form method="POST">
        {% csrf_token %}
        <table>
            {{ form.as_table }}
        </table>
        <button type = "submit">작성</button>
    </form>
{% endblock %}
```

동일하게
HTTP method 가 POST 방식으로
Form 을 처리하는 것을
알 수 있어요

〈 CR _ use form 〉

```
{% extends 'index.html' %}

{% block title %}
    <title>Create</title>
{% endblock %}

{% block content %}
    <h1>Create Page</h1>
    <hr>
    <form method="POST">
        {% csrf_token %}
        <table>
            {{ form.as_table }}
        </table>
        <button type = "submit">작성</button>
    </form>
{% endblock %}
```

동일하게
Csrf_token 으로 form 을
검증합니다.

이걸 작성하지 않으면
웹 사이트의 보안에
문제가 생길 여지가 있겠죠?

〈 CR _ use form 〉

```
{% extends 'index.html' %}

{% block title %}
    <title>Create</title>
{% endblock %}

{% block content %}
    <h1>Create Page</h1>
    <hr>
    <form method="POST">
        {% csrf_token %}
        <table>
            {{ form.as_table }}
        </table>
        <button type = "submit">작성</button>
    </form>
{% endblock %}
```

이는 form 을
.as_table

즉, table 과 같은 형식으로
가져오겠다는 코드입니다 ~

{ CR _ use form }

crudapp > 📁 views.py > ⚒ create

crudapp / views.py

```
1  from django.shortcuts import render, redirect
2  from .forms import BlogForm
3  from django.utils import timezone
4
5  # Create your views here.
6
7  def index(request):
8      return render(request, 'index.html')
9
10 def create(request):
11     if request.method == 'POST':
12         form = BlogForm(request.POST)
13         if form.is_valid():
14             form = form.save(commit = False)
15             form.pub_date = timezone.now()
16             form.save()
17             return redirect('read')
18     else:
19         form = BlogForm
20         return render(request, 'create.html', {'form': form})
```

작성해주세요

〈CR_use form〉

crudapp / views.py

```
10 def create(request):
11     if request.method == 'POST':
12         form = BlogForm(request.POST)
13         if form.is_valid():
14             form = form.save(commit = False)
15             form.pub_date = timezone.now()
16             form.save()
17             return redirect('read')
18     else:
19         form = BlogForm
20     return render(request, 'create.html', {'form': form})
```

request.method == 'POST'
즉, 정보를 받는 방식이 POST라면
아래 코드가 실행되겠죠?

〈CR – use form〉

crudapp / views.py

```
10 def create(request):
11     if request.method == 'POST':
12         form = BlogForm(request.POST)
13         if form.is_valid():
14             form = form.save(commit = False)
15             form.pub_date = timezone.now()
16             form.save()
17             return redirect('read')
18     else:
19         form = BlogForm
20     return render(request, 'create.html', {'form': form})
```

정보를 받는 방식이 POST 가 아닌 경우
즉, GET 과 같은 경우에는
아래 코드가 실행될거에요
이 경우는 Form 을 보여줄거에요

〈CR – use form〉

crudapp / views.py

```
10 def create(request):
11     if request.method == 'POST':
12         form = BlogForm(request.POST)
13         if form.is_valid():
14             form = form.save(commit = False)
15             form.pub_date = timezone.now()
16             form.save()
17             return redirect('read')
18     else:
19         form = BlogForm
20     return render(request, 'create.html', {'form': form})
```

forms.py에서 import 해온 BlogForm을
form이라는 변수에 저장한 후
POST로 요청받은 폼이 정보를 잘 받았는지 확인해줘요

〈CR_use form〉

crudapp / views.py

```
10 def create(request):
11     if request.method == 'POST':
12         form = BlogForm(request.POST)
13         if form.is_valid():
14             form = form.save(commit = False)
15             form.pub_date = timezone.now()
16             form.save()
17             return redirect('read')
18     else:
19         form = BlogForm
20     return render(request, 'create.html', {'form': form})
```

BlogForm 이 유효하다면
즉, 요청을 잘 받았다면

〈CR_use form〉

crudapp / views.py

```
10 def create(request):
11     if request.method == 'POST':
12         form = BlogForm(request.POST)
13         if form.is_valid():
14             form = form.save(commit = False)
15             form.pub_date = timezone.now()
16             form.save()
17             return redirect('read')
18     else:
19         form = BlogForm
20     return render(request, 'create.html', {'form': form})
```

Form 을 잠시 저장 대기 상태에 두고

〈CR_use form〉

crudapp / views.py

```
10 def create(request):
11     if request.method == 'POST':
12         form = BlogForm(request.POST)
13         if form.is_valid():
14             form = form.save(commit = False)
15             form.pub_date = timezone.now()
16             form.save()
17             return redirect('read')
18     else:
19         form = BlogForm
20     return render(request, 'create.html', {'form': form})
```

현재 날짜 및 시간을 pub_date field에 저장해줘요

{ CR _ use form }

crudapp / views.py

```
10 def create(request):
11     if request.method == 'POST':
12         form = BlogForm(request.POST)
13         if form.is_valid():
14             form = form.save(commit = False)
15             form.pub_date = timezone.now()
16             form.save()
17             return redirect('read')
18     else:
19         form = BlogForm
20     return render(request, 'create.html', {'form': form})
```

Form 을 정말 전체 저장해주고

〈CR_use form〉

crudapp / views.py

```
10 def create(request):
11     if request.method == 'POST':
12         form = BlogForm(request.POST)
13         if form.is_valid():
14             form = form.save(commit = False)
15             form.pub_date = timezone.now()
16             form.save()
17             return redirect('read')
18     else:
19         form = BlogForm
20     return render(request, 'create.html', {'form': form})
```

‘read’로 redirect 해줄게요
Redirect는 아까 배웠죠?

〈CR – use form〉

crud2project > urls.py > ...

crud2project / urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 import crudapp.views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', crudapp.views.index, name = 'index'),
23     path('create/', crudapp.views.create, name = 'create'),
24 ]
25
```

요로케 작성해주세요

CRUD

〈 CR _ use form 〉

crudapp >  views.py >  read

crudapp / views.py

```
1  from django.shortcuts import render, redirect  
2  from .forms import BlogForm  
3  from django.utils import timezone  
4  from .models import Blog
```

Import 해주시구요

```
22  
23  def read(request):  
24      blogs = Blog.objects.all()  
25      return render(request, 'read.html', {'blogs': blogs})
```

create 함수 밑에 이어서 작성해주세요

{ CR _ use form }

crudapp / templates / read.html

CRUD2

- crud2project
 - __pycache__
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py
- crudapp
 - __pycache__
 - migrations
 - templates
 - create.html
 - index.html
 - read.html

crudapp > templates > <> read.html > ...

```
1  {% extends 'index.html' %}  
2  
3  {% block title %}  
4      <title>Read</title>  
5  {% endblock %}  
6  
7  {% block content %}  
8      <h1>Read Page</h1>  
9      <hr>  
10     {% for blogs in blogs %}  
11         <h3>{{ blogs.title }}</h3>  
12         <p>{{ blogs.pub_date }}</p>  
13         <p>{{ blogs.body }}</p>  
14     {% endfor %}  
15  {% endblock %}
```

crudapp / templates 안에 read.html 을 만들고 작성해주세요

〈CR – use form〉

crud2project > urls.py > ...

crud2project / urls.py

```
16  from django.contrib import admin
17  from django.urls import path
18  import crudapp.views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', crudapp.views.index, name = 'index'),
23     path('create/', crudapp.views.create, name = 'create'),
24     path('read/', crudapp.views.read, name = 'read'),
25 ]
26
```

urls.py에 url 연결해주시고

{ CR _ use form }

crudapp > templates > index.html > html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      {% block title %}
8          <title>Index</title>
9      {% endblock %}
10 </head>
11 <body>
12     <nav>
13         <a href="{% url 'create' %}">글 작성하기</a>
14         <a href="{% url 'read' %}">글 모아보기</a>
15     </nav>
16     {% block content %}
17         <h1>Index Page</h1>
18         <hr>
19     {% endblock %}
20 </body>
21 </html>
```

crudapp / templates / index.html

index.html에 navbar를
만들어줄게요

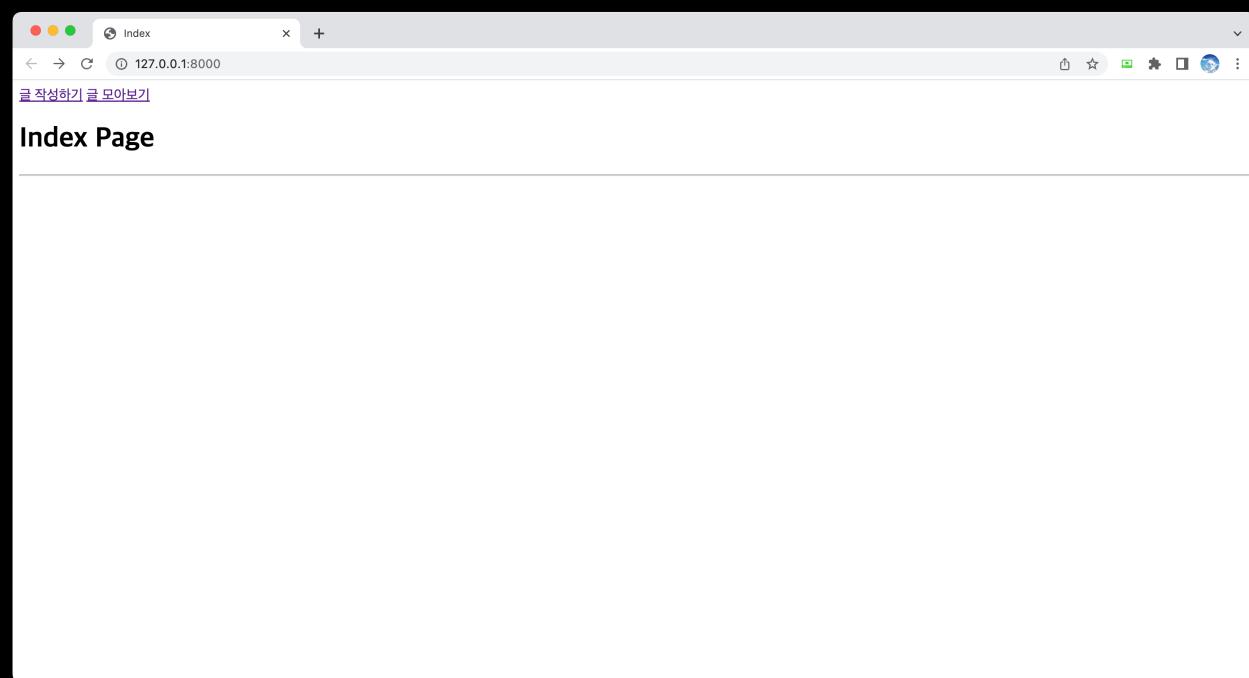
〈 CR – use form 〉

Windows :: python manage.py runserver

Mac :: python3 manage.py runserver

런서버 해줄게요

〈 CR – use form 〉



런서버 하면 이렇게 뜨시죠?
nav바를 눌러서 오류없이
이동이 잘 되는지 봄주세요

A screenshot of a web browser window titled "Create". The address bar shows "127.0.0.1:8000/create/". Below the address bar, there is a link labeled "글 작성하기 글 모아보기". The main content area is labeled "Create Page". It contains two input fields: "Title:" and "Body:", both represented by empty text boxes. At the bottom right of the form is a button labeled "작성".

create 페이지

A screenshot of a web browser window titled "Read". The address bar shows "127.0.0.1:8000/read/". Below the address bar, there is a link labeled "글 작성하기 글 모아보기". The main content area is labeled "Read Page".

read 페이지

〈 CR – use form 〉

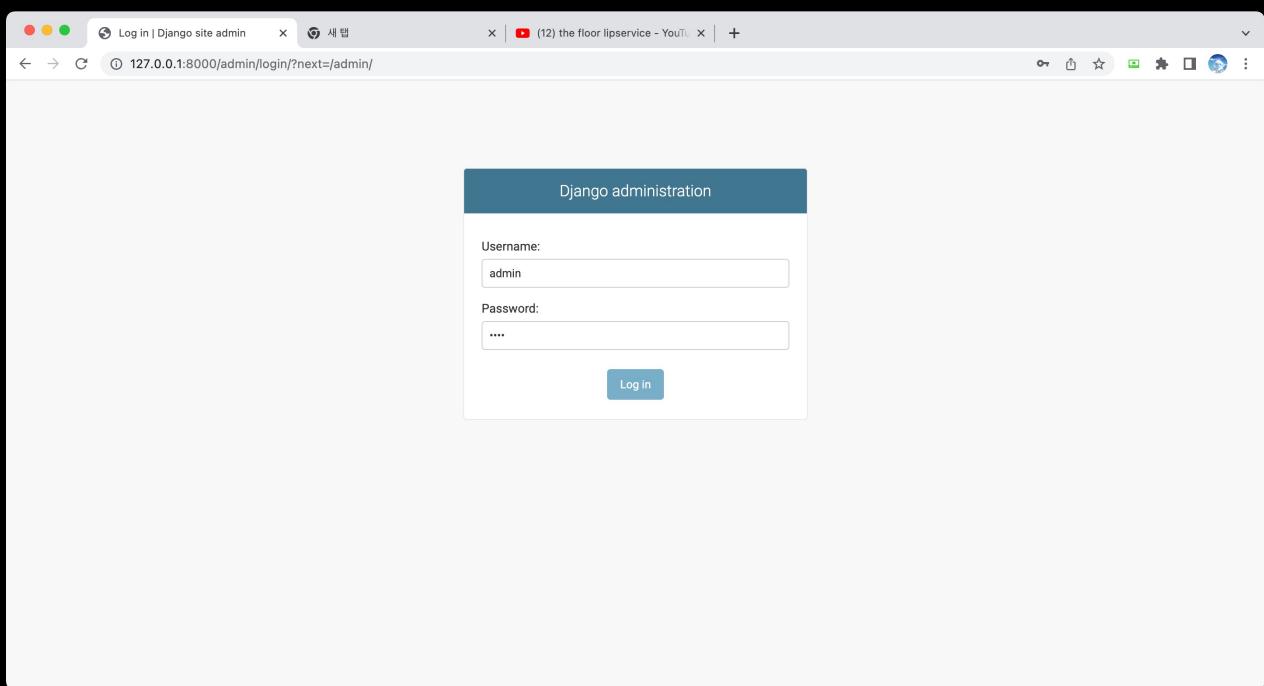
The screenshot shows a 'Create Page' form on a web page. The title is 'Create Page'. There are two input fields: 'Title' containing '안녕하세요' and 'Body' containing the Korean text '첫 글을 작성합니다. ㅎ^. ^', '오늘의 작업곡은 The Floor 「리ップ서비스」'. Below the fields is a '작성' button.

The screenshot shows a 'Read Page' with the title '안녕하세요'. It lists three posts with their titles, dates, and brief descriptions. The first post is 'May 22, 2023, 5:31 a.m.' with '첫 글을 작성합니다. ㅎ^. ^, 오늘의 작업곡은 The Floor 「리ップ서비스」'. The second post is 'May 22, 2023, 5:32 a.m.' with '하나 더 작성할 테니까'. The third post is 'May 22, 2023, 5:32 a.m.' with '오류가 빗기 때문이지.. 눈물이 나기 때문이지...'. Below the posts is the text '근데 잘댐 기본조아짐'.

create 페이지에 글을 작성해주고
작성 버튼을 눌러줄게요

이렇게 잘 되시나요?
잘 되셔야 … 해요 …

〈 CR – use form 〉



주소창에
<http://127.0.0.1:8000> 뒤에
/admin 작성하시구
로그인 해주세요

A screenshot of the Django admin interface showing the "Blogs" list. The title bar says "Select blog to change | Django site admin". The left sidebar shows "Groups", "Users", and "CRUDAPP" sections, with "Blogs" highlighted. The main area shows a table with four rows: "BLOG", "첨바 첨바", "근데 절대 기본조아침", "하나 더 작성한 웨나면", and "안녕하세요". An orange box highlights the "Blogs" section in the sidebar. Below the table, it says "4 blogs". A large white box with black text "이거 누르고 들어가주세요" is overlaid on the top right of the screenshot.

각 글을 눌러서 내용도 확인해주세요!

〈 CR – use form 〉

잘 되신다면

Windows :: ctrl + c

Mac :: control + c

서버 종료 해주세요

〈 CR _ use form 〉

crudapp >  views.py >  detail

crudapp / views.py

```
1  from django.shortcuts import render, redirect, get_object_or_404
2  from .forms import BlogForm
3  from django.utils import timezone
4  from .models import Blog
```

Import 해주시고

```
25  return render(request, 'crudapp/detail.html', {'blogs': blogs})
26
27 def detail(request, id):
28     blog = get_object_or_404(Blog, id = id)
29     return render(request, 'detail.html', {'blog': blog})
```

read 함수에 이어 코드를 작성해주세요

〈 CR _ use form 〉

crudapp > views.py > detail

crudapp / views.py

```
1  from django.shortcuts import render, redirect, get_object_or_404
2  from .forms import BlogForm
3  from django.utils import timezone
4  from .models import Blog
```

Import 해주시고

```
25  return render(request, 'crudapp/detail.html', {'blogs': blogs})
26
27 def detail(request, id):
28     blog = get_object_or_404(Blog, id = id)
29     return render(request, 'detail.html', {'blog': blog})
```

read 함수에 이어 코드를 작성해주세요

〈 CR _ use form 〉

crudapp > templates > detail.html

```
1  {% extends 'index.html' %}\n2\n3  {% block title %}\n4      <title>Detail</title>\n5  {% endblock %}\n6\n7  {% block content %}\n8      <h1>Detail Page</h1>\n9      <hr>\n10     <h3>{{ blog.title }}</h3>\n11     <p>{{ blog.pub_date }}</p>\n12     <p>{{ blog.body }}</p>\n13  {% endblock %}
```

crudapp / templates 에 detail.html 을 생성해주고
코드를 작성해주세요

{ CR - use form }

crudapp / templates / read.html

crudapp > templates > <> read.html > h3 > a

```
1  {% extends 'index.html' %}  
2  
3  {% block title %}  
4      <title>Read</title>  
5  {% endblock %}  
6  
7  {% block content %}  
8      <h1>Read Page</h1>  
9      <hr>  
10     {% for blogs in blogs %}  
11         <h3><a href="{% url 'detail' blogs.id %}">{{ blogs.title }}</a></h3>  
12         <p>{{ blogs.pub_date }}</p>  
13         <p>{{ blogs.body }}</p>  
14     {% endfor %}  
15  {% endblock %}
```

read.html 의 코드를 수정해주세요

〈 CR – use form 〉

crud2project > urls.py > ...

```
16     from django.contrib import admin
17     from django.urls import path
18     import crudapp.views
19
20     urlpatterns = [
21         path('admin/', admin.site.urls),
22         path('', crudapp.views.index, name = 'index'),
23         path('create/', crudapp.views.create, name = 'create'),
24         path('read/', crudapp.views.read, name = 'read'),
25         path('detail/<str:id>/', crudapp.views.detail, name = 'detail'),
26     ]
27
```

crud2project / urls.py 에 detail url 연결을 해주세요.

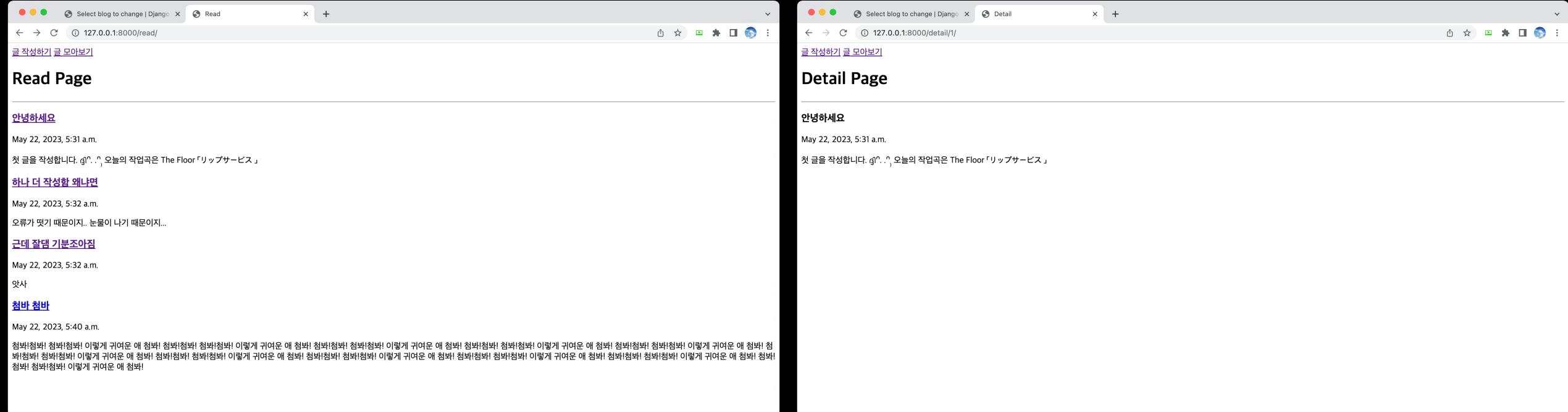
〈 CR – use form 〉

Windows :: python manage.py runserver

Mac :: python3 manage.py runserver

다시 런서버 해줄게요

〈 CR – use form 〉



Read 페이지로 이동해주고
작성해준 글을 클릭해주세요

이렇게 detail 페이지로
이동이 잘 되시죠?

〈 CR – use form 〉

잘 되신다면

Windows :: ctrl + c

Mac :: control + c

서버 종료!

긴 강의 들느라 너무 수고하셨어요

오늘 쓴 crud1 폴더와 crud2 폴더는
다음 실습에도 사용할 예정입니다 ㅋㅋ..ㅋ

그러니 오류 없이 복습 해오셔야
다음 수업시간에 수월하게 실습할 수 있겠죠 ?