

Python Django 게시판 (MariaDB)

1단계 – 장고설치, 프로젝트, 앱생성

환경 준비

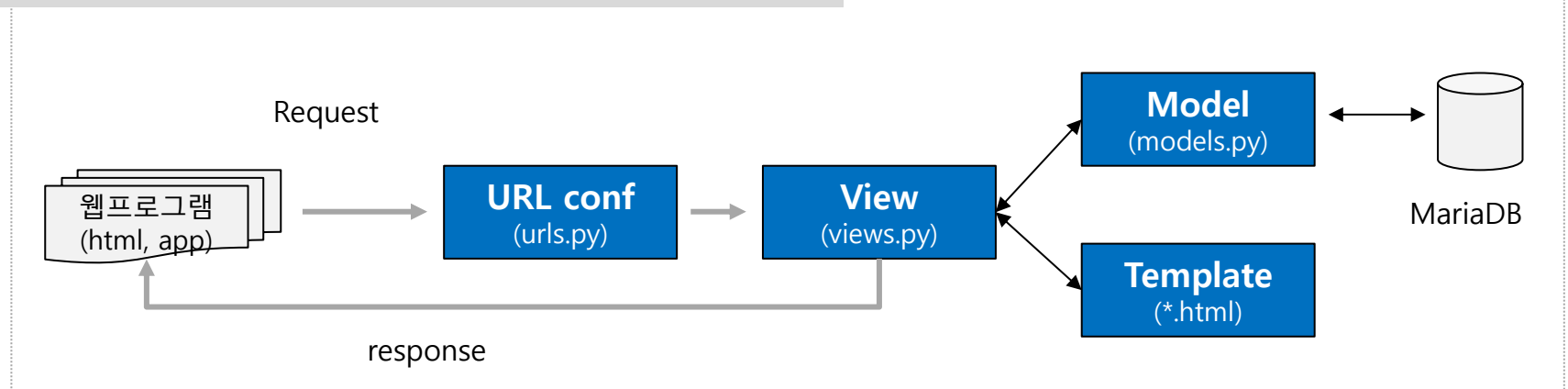
- Python 설치, 가상환경 구성
- MariaDB 설치
- 장고(Django) 설치

✓ **django-admin startproject config .**

- Python --version
- django-admin --version
- pip install virtualenvwrapper-win
- mkvirtualenv test1
- pip install django
- django-admin --version
- pip install mysqlclient

<https://www.youtube.com/watch?v=OTmQQjsl0eg>

MVT 패턴 (일반적인 MVC패턴과 거의 동일)



설계 - 화면

게시판

등록

1

[게시판 입니다...](#)

홍길동

5

삭제

보기

2

[게시판 입니다...](#)

고길동

3

삭제

보기

3

[게시판 입니다...](#)

이길동

1

삭제

보기

게시판 - 등록

제목 :

작성자 :

저장

게시판 - 보기

목록

제목 :

게시판 입니다...

작성자 :

홍길동

내용입니다.

내용입니다.

내용입니다.

수정

게시판 - 수정

제목 :

게시판 입니다...

작성자 :

홍길동

내용입니다.

내용입니다.

내용입니다.

저장

구축 (MariaDB)

HeidiSQL을 사용하여 board 테이블 만들기

기본

옵션

인덱스

외래 키

분할

CREATE 코드

ALTER 코드

이름:

board

코멘트:

열:

추가

제거

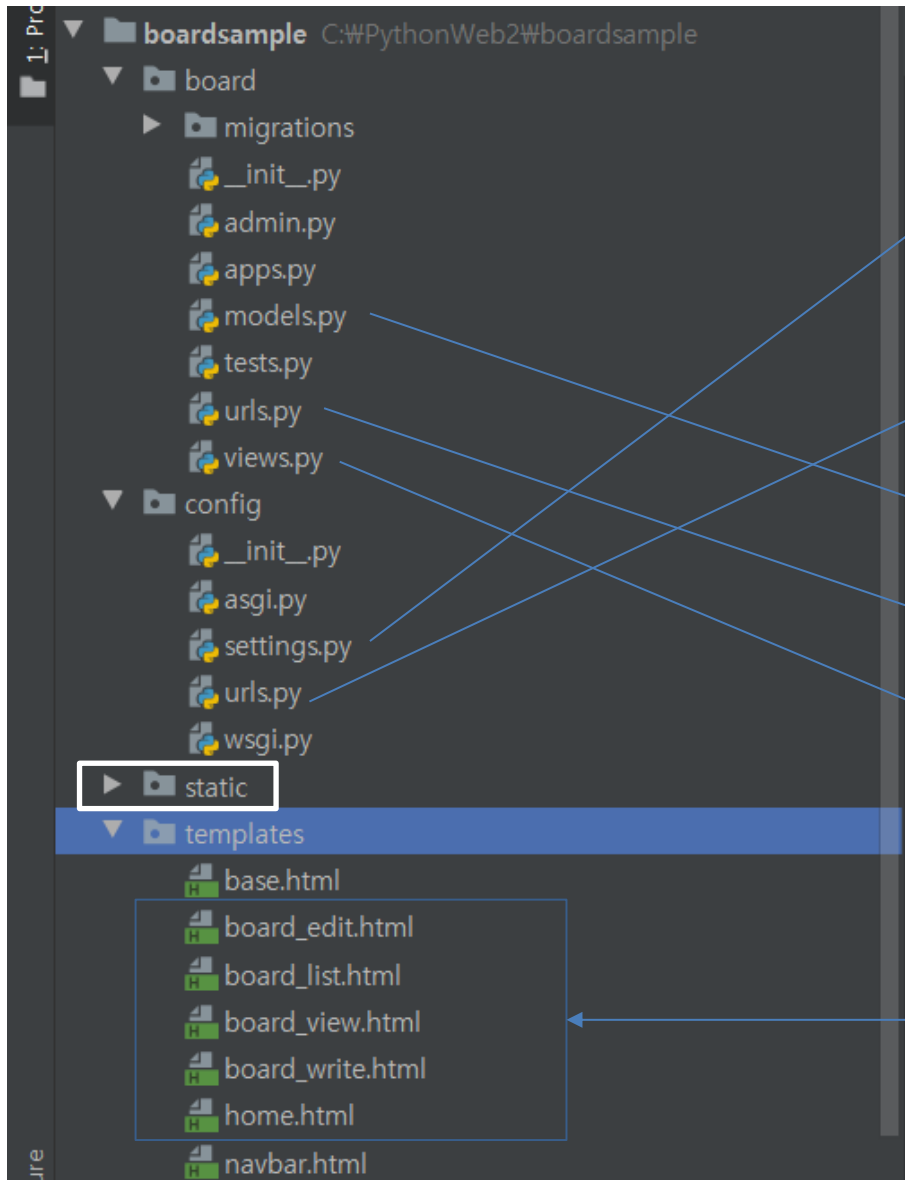
위로

아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로...	기본값
1	b_no	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	b_title	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
3	b_note	VARCHAR	4096	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
4	parent_no	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0
5	b_writer	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
6	b_count	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
7	b_date	DATETIME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

Board 설계

python manage.py startapp board



DATABASES
INSTALLED_APPS
TEMPLATES
STATIC_URL

```
from django.urls import path
from django.conf.urls import include

urlpatterns = [
    path("", include('board.urls')),
]
```

class Board 정의

Model
(models.py)

url정의

함수 정의 및 render

View
(views.py)

Template
(* .html)

/static/

config/settings.py

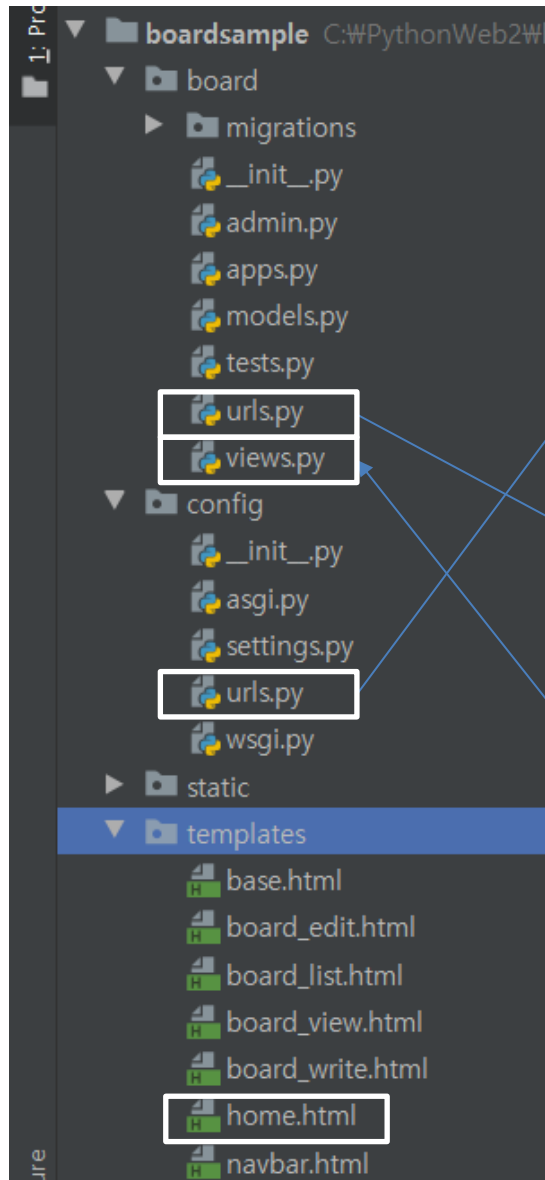
```
INSTALLED_APPS = [  
    'board',  
]
```

```
TEMPLATES = [  
    'DIRS': [os.path.join(BASE_DIR, 'templates')],  
]
```

```
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR, 'static'),  
]
```

2단계 – Template분리, Bootstrap 디자인 적용

Urls.py



```
from django.contrib import admin
from django.urls import path
from django.conf.urls import url, include
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('board.urls')),
]
```

```
from django.conf.urls import url, include
from django.urls import path
```

```
from . import views
```

```
urlpatterns = [
    path("", views.home, name="home"),
]
```

<http://127.0.0.1:8000/> 로 접속하게 되면
views.py 에 있는 home 함수를 호출한다.

```
from django.shortcuts import render
def home(request):
    return render(request, "home.html")
```

Template (.html)

```
<!DOCTYPE html>
<html lang="en">

  <head>
    <title>게시판 예제입니다. </title>
    <style>
      .myclass {
        color:blue;
      }
    </style>
    <script LANGUAGE="javascript">
      function testFunction() {
        alert("Javascript function...");
      }
    </script>
  </head>

  <body>
    <div>
      <h3>장고(Django) + MariaDB 게시판</h3>
      <ul>
        <li>HOME</li>
        <li>Call function </li>
        <li>COP</li>
      </ul>
    </div>
  </body>

</html>
```

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>
      {% block title %}
      게시판 예제입니다.
      {% endblock %}
    </title>
    {% block styles %}
    <style>
      .myclass {
        color:blue;
      }
    </style>
    {% endblock %}
    {% block scripts %}
    <script LANGUAGE="javascript">
      function testFunction() {
        alert("Javascript function...");
      }
    </script>
    {% endblock %}
  </head>
  <body>
    {% block body %}
    <div>
      <h3>장고(Django) + MariaDB 게시판</h3>
      <ul>
        <li>HOME</li>
        <li>Call function </li>
        <li>COP </li>
      </ul>
    </div>
    {% endblock %}
  </body>
</html>
```

Template (.html) 분리

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>
      {% block title %}
      게시판 예제입니다.
      {% endblock %}
    </title>
    {% block styles %}
    <style>
    </style>
    {% endblock %}
    {% block scripts %}
    <script LANGUAGE="javascript">
    </script>
    {% endblock %}
  </head>
  <body>
    {% block body %}
    <div>
      <h3>장고(Django) + MariaDB 게시판</h3>
      <ul>
        <li>HOME</li>
        <li>Call function </li>
        <li>COP </li>
      </ul>
    </div>
    {% endblock %}
  </body>
</html>
```

{% extends "base.html" %}

{% include "navbar.html" %}

base.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>
      {% block title %}
      게시판 예제입니다.
      {% endblock %}
    </title>
    {% block styles %}
    <style>
    </style>
    {% endblock %}
    {% block scripts %}
    <script LANGUAGE="javascript">
    </script>
    {% endblock %}
  </head>
  <body>
```

navbar.html

```
<div>
  <h3>장고(Django) + MariaDB 게시판</h3>
  <ul>
    <li>HOME</li>
    <li>Call function </li>
    <li>COP </li>
  </ul>
</div>
```

{{ block.super }}

<!-- 부모 템플릿에서 블록의 내용을 가져와야 할 경우 -->

example.html

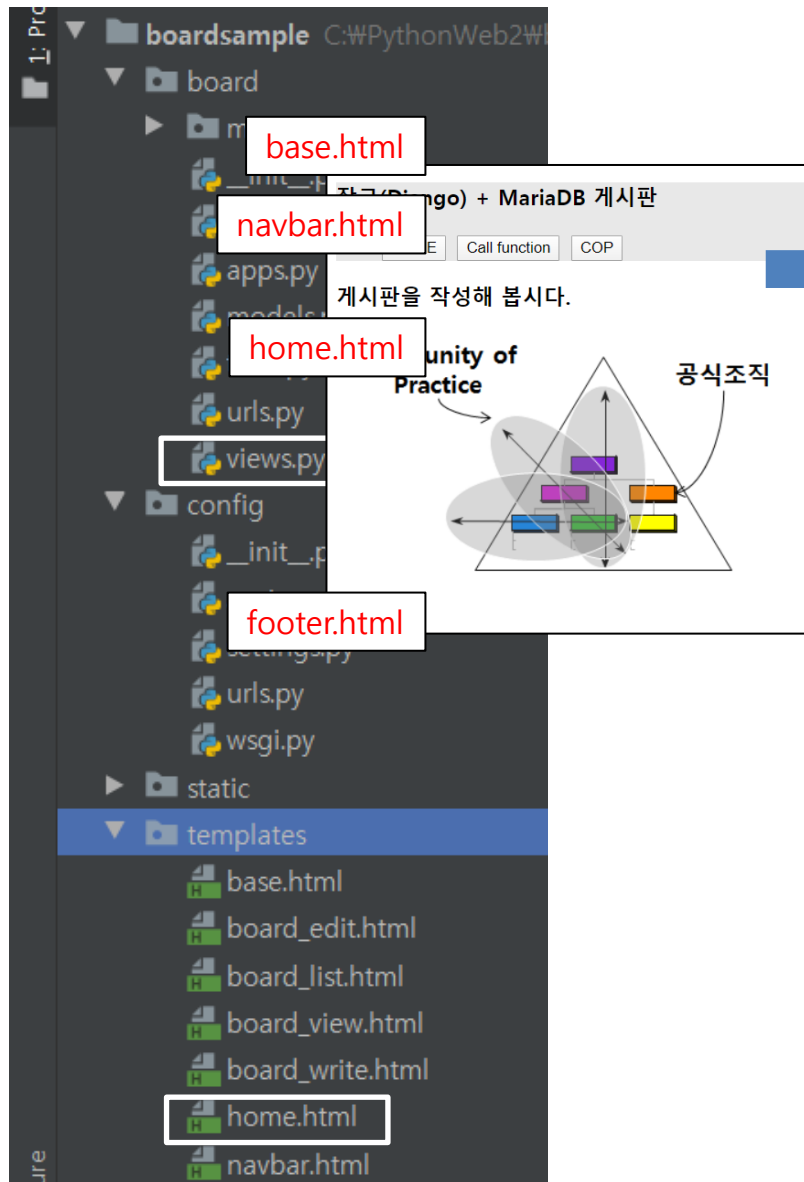
```
{% block content %}  
    <p>Some example text.</p>  
{% endblock %}
```

```
{% extends 'example.html' %}  
  
{% block content %}  
    <p>The previous example text is replaced by this text.</p>  
{% endblock %}
```

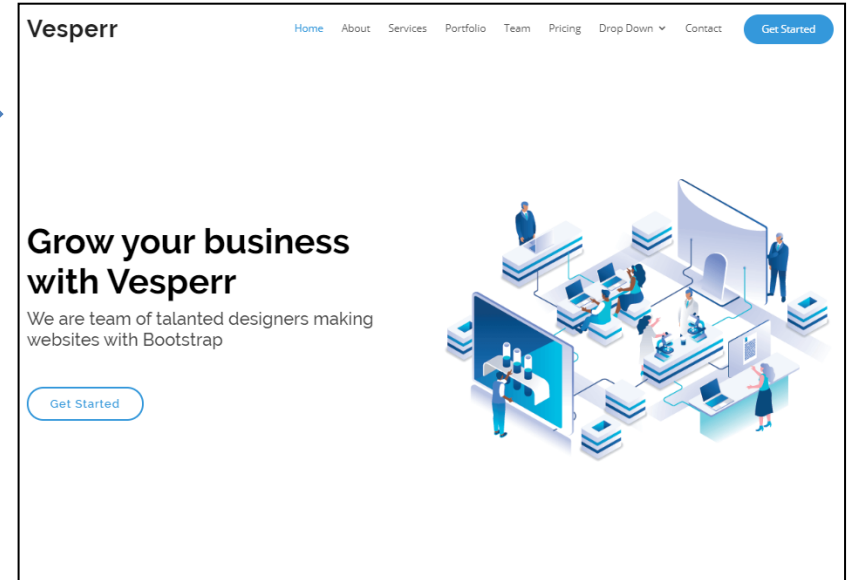
```
{% extends 'example.html' %}  
  
{% block content %}  
    {{ block.super }}  
    <p>The previous example text is replaced by this text.</p>  
{% endblock %}
```

디자인을 입혀 봅시다.

구 화면



bootstrapmade.com



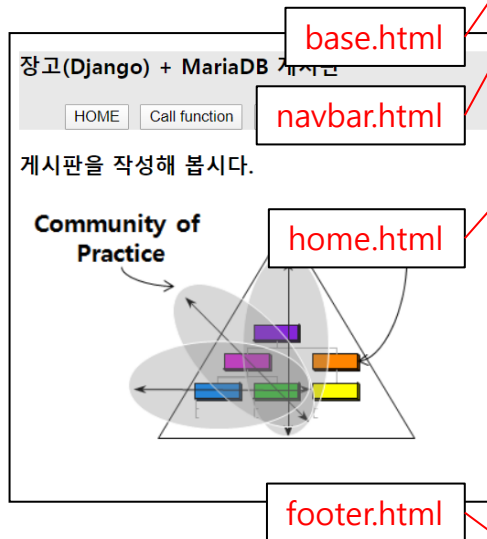
Bootstrap 적용

구 화면

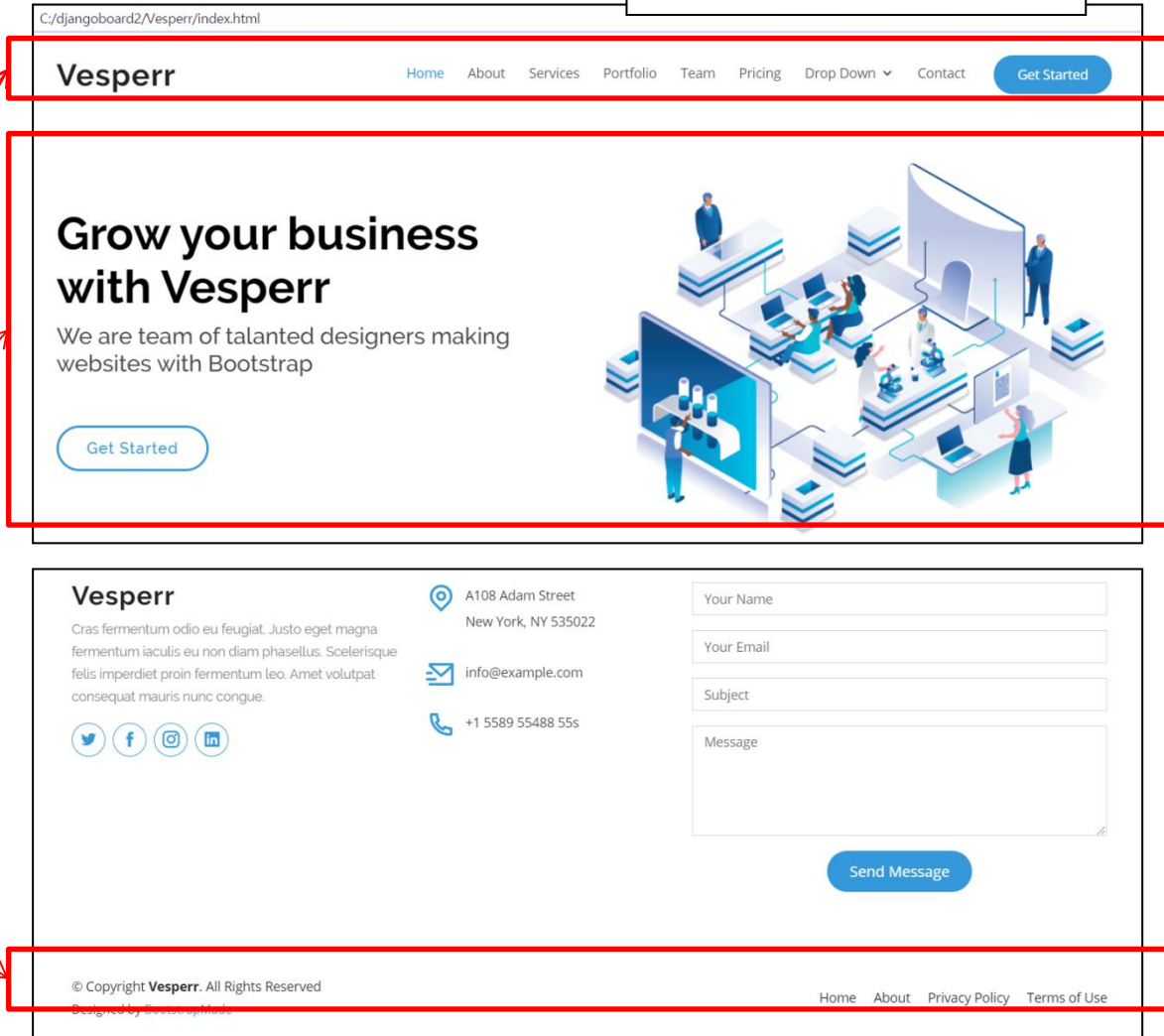


bootstrapmade.com

1. 디자인 템플릿 다운로드
2. static으로 복사
3. Index.html등에서 선택 적용



- assets
- forms
- changelog.txt
- index.html
- index2.html
- portfolio-details.html
- Readme.txt

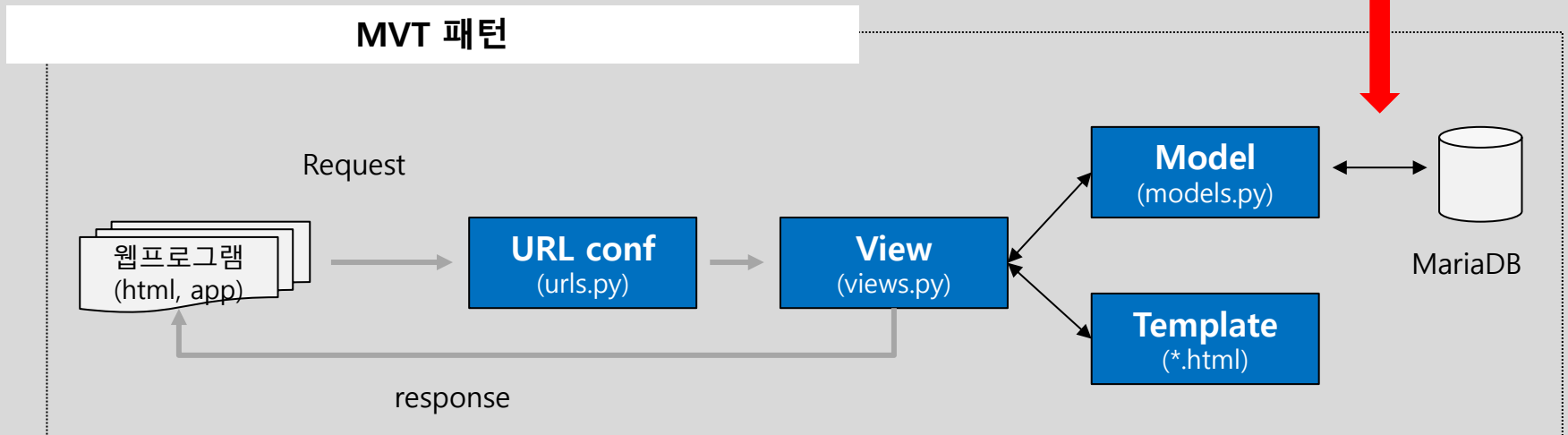


3단계 - 게시판

SQLAlchemy

ORM
Object Relational Mapper

MVT 패턴

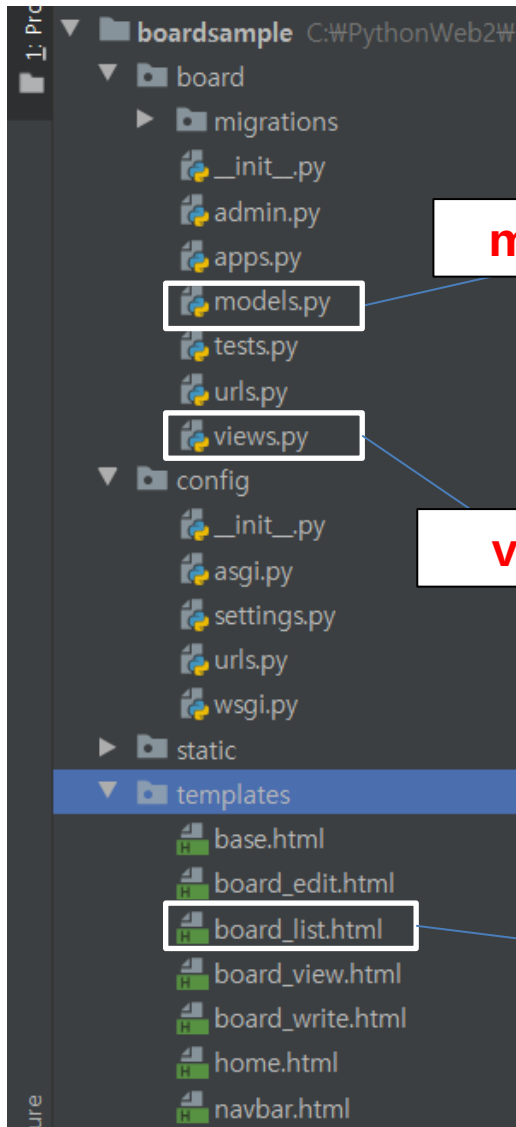


config/settings.py

```
pip install mysqlclient
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'edudb',  
        'USER': 'root',  
        'PASSWORD': 'intra165',  
        'HOST': '127.0.0.1',  
        'PORT': '3306',  
    }  
}
```


게시판 목록 (조회)



models.py

views.py

board 테이블

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로...	기본값
1	b_no	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	b_title	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
3	b_note	VARCHAR	4096	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
4	parent_no	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0
5	b_writer	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
6	b_count	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
7	b_date	DATETIME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

b_no	b_title	b_note	b_writer
1	제목입니다.	내용입니다	홍길동
2	제목입니다.2	내용입니다2	고길동
			김길동

rsBoard 레코드셋을 만들어서
템플릿(board_list.html)에 전달

Html로 표현

장고(Django) + MariaDB 게시판 예제

HOME

BOARD

게시판 목록

INSERT

5	게시판2 제목입니다~~	게시판 내용입니다~~	마이콜~~	Dec. 28, 2019, midnight	보기	삭제	Ajax삭제
6	게시판.. 제목입니다	게시판 내용입니다	크리콜	Dec. 28, 2019, midnight	보기	삭제	Ajax삭제
8	제목 수정입니다.			Dec. 28, 2019, midnight	보기	삭제	Ajax삭제
11	sfdasfaf			None	보기	삭제	Ajax삭제
14	1			April 22, 2020, midnight	보기	삭제	Ajax삭제

게시판 Table과 ORM

기본 옵션 인덱스 외래 키 분할 </> CREATE 코드 </> ALTER 코드								
이름:		board						
코멘트:								
열: 추가 제거 위로 아래로								
#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로...	기본값	
1	b_no	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT	
2	b_title	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
3	b_note	VARCHAR	4096	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
4	parent_no	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	
5	b_writer	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
6	b_count	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	
7	b_date	DATETIME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	



models.py

```
class Board(models.Model):  
    b_no = models.AutoField(db_column='b_no', primary_key=True)  
    b_title = models.CharField(db_column='b_title', max_length=255)  
    b_note = models.TextField(db_column='b_note', )  
    b_writer = models.CharField(db_column='b_writer', max_length=50)  
    parent_no = models.IntegerField(db_column='parent_no', default=0)  
    b_count = models.IntegerField(db_column='b_count', default=0)  
    b_date = models.DateTimeField(db_column='b_date', )
```

마이그레이션

python manage.py makemigrations

python manage.py migrate

makemigrations : models.py에서 적용한 변경사항이나 추가된 혹은 삭제된 사항들을 감지하여 파일로 생성

migrate : 적용되지 않은 migrations들을(설정값 들을) 적용시키는 역할

The Commands

There are several commands which you will use to interact with migrations and Django's handling of database schema:

- **migrate**, which is responsible for applying and unapplying migrations.
- **makemigrations**, which is responsible for creating new migrations based on the changes you have made to your models.
- **sqlmigrate**, which displays the SQL statements for a migration.
- **showmigrations**, which lists a project's migrations and their status.

You should think of migrations as a version control system for your database schema. **makemigrations** is responsible for packaging up your model changes into individual migration files - analogous to commits - and **migrate** is responsible for applying those to your database.

설계 - 화면

Board → board_list.html

게시판

등록

1	게시판 입니다...	홍길동	5	삭제	보기
2	게시판 입니다...	고길동	3	삭제	보기
3	게시판 입니다...	이길동	1	삭제	보기

board_delete

board_write → board_write.html → board_insert

게시판 - 등록

제목 :

작성자 :

저장

board_view → board_view.html

게시판 - 보기

목록

제목 : 게시판 입니다...

작성자 : 홍길동

내용입니다.
내용입니다.
내용입니다.

수정

board_edit → board_edit.html → board_update

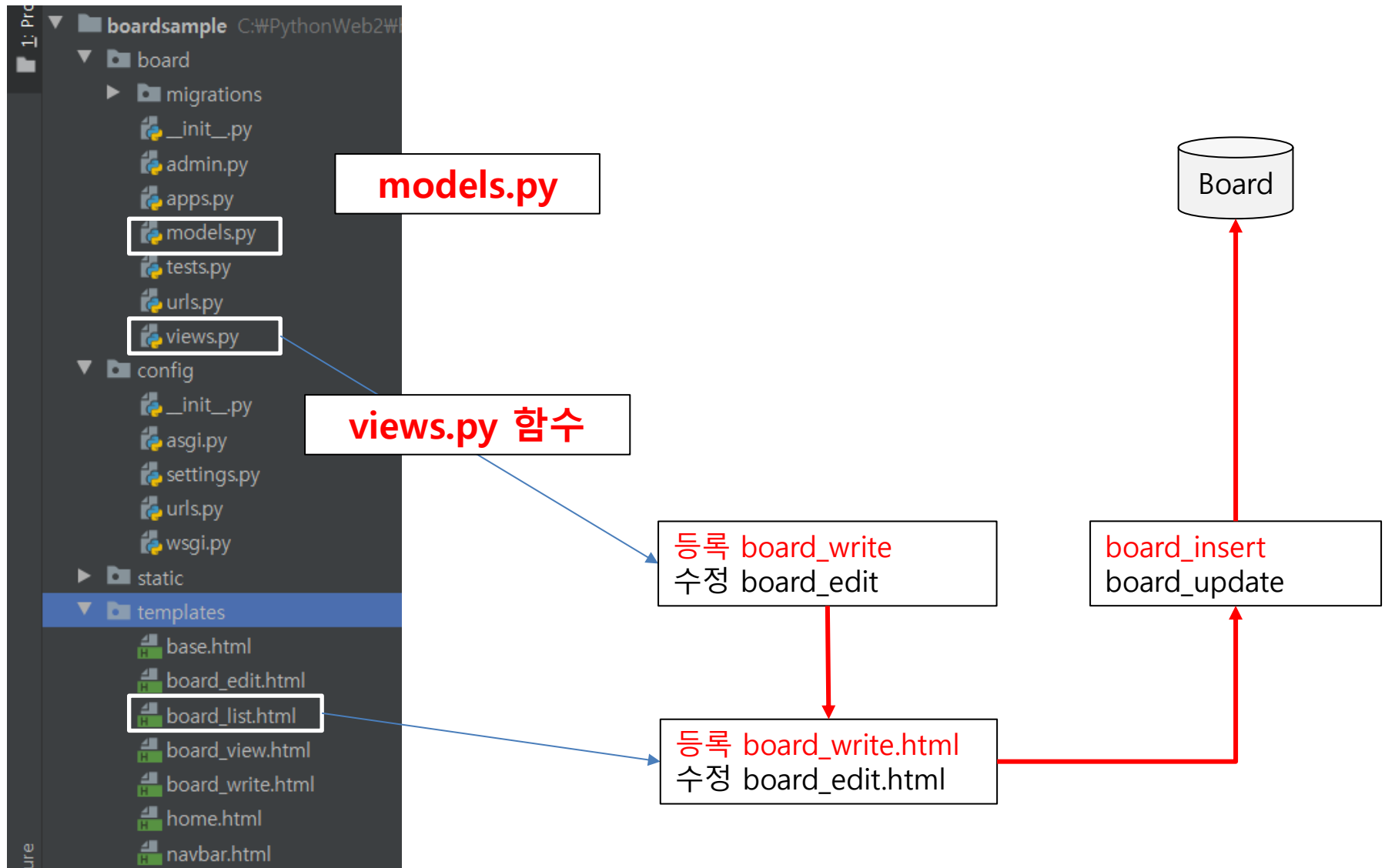
게시판 - 수정

제목 :

작성자 :

저장

게시판 등록,수정,삭제



심 화

- ✓ 링크 구성해 보기
- ✓ 댓글
- ✓ Page scroll 자동으로 조회... (혹은 페이지 표기)
- ✓ 이미지/파일 첨부
- ✓ 에러 체크
- ✓ **Ajax로 처리 (get/post 방식과 비교)**
- ✓ Ajax로 count 증가
- ✓ Css와 image등 resource처리
- ✓ 특수문자 처리 - get방식에서 &,'"' 등 처리
- ✓ ANSI/UTF8, encode/decode

- ✓ 파일 첨부