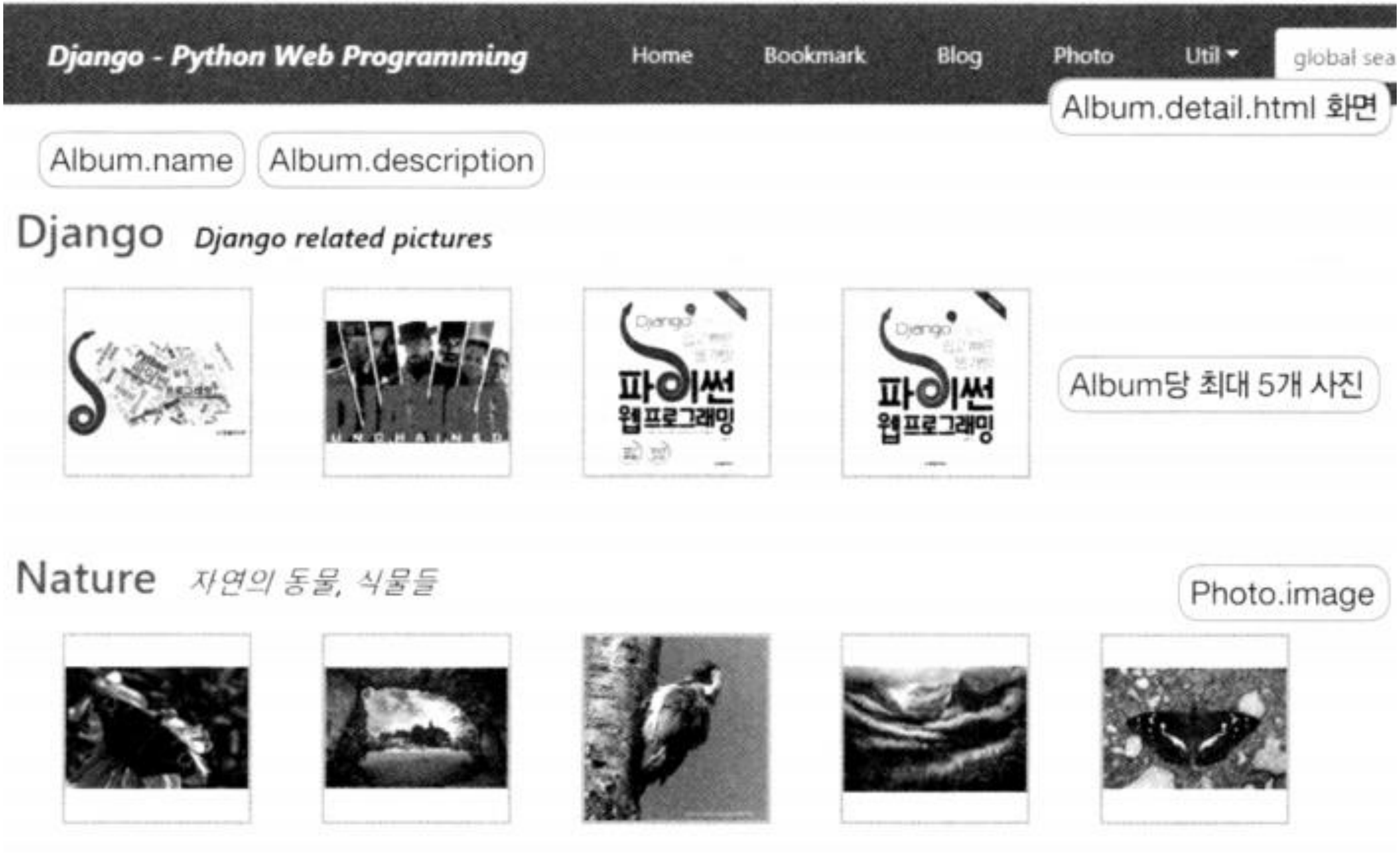
실전 프로그램 개발
- Photo 앱 -

애플리케이션 설계하기

❖ 화면 설계



Album.detail.html 화면

Album.name

Album.description

Django Django related pictures

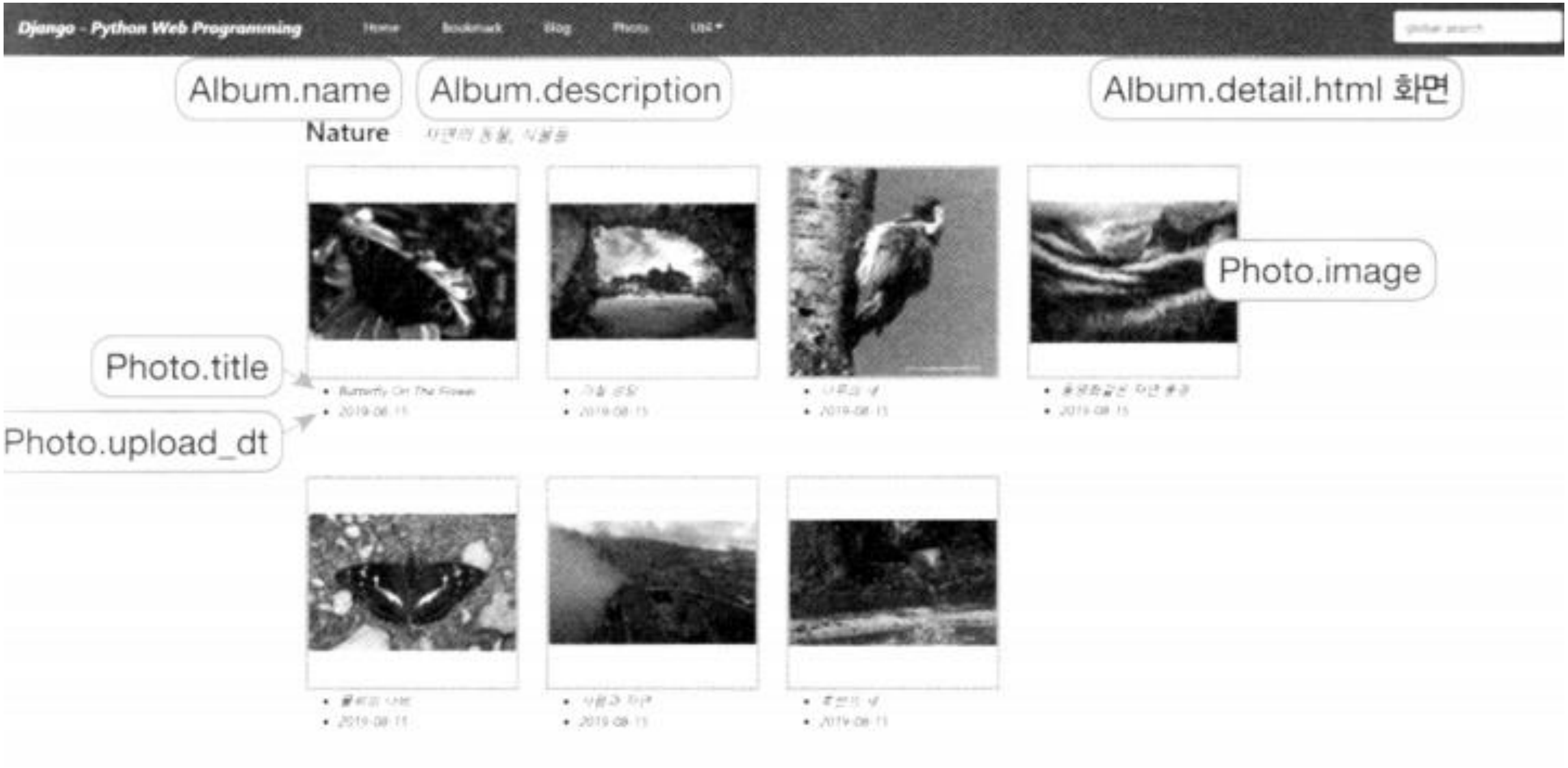
Album당 최대 5개 사진

Nature 자연의 동물, 식물들

Photo.image

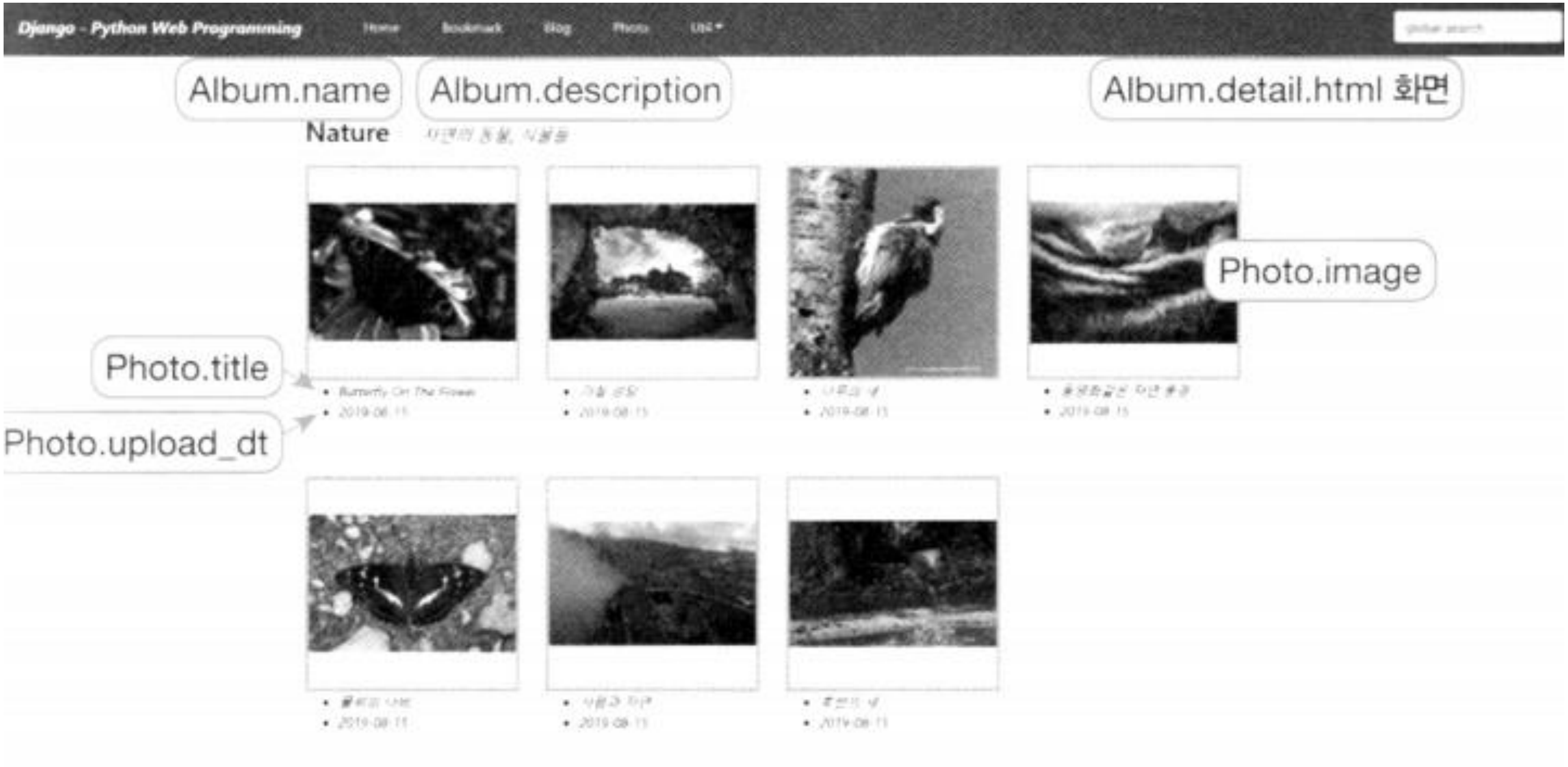
애플리케이션 설계하기

❖ 화면 설계



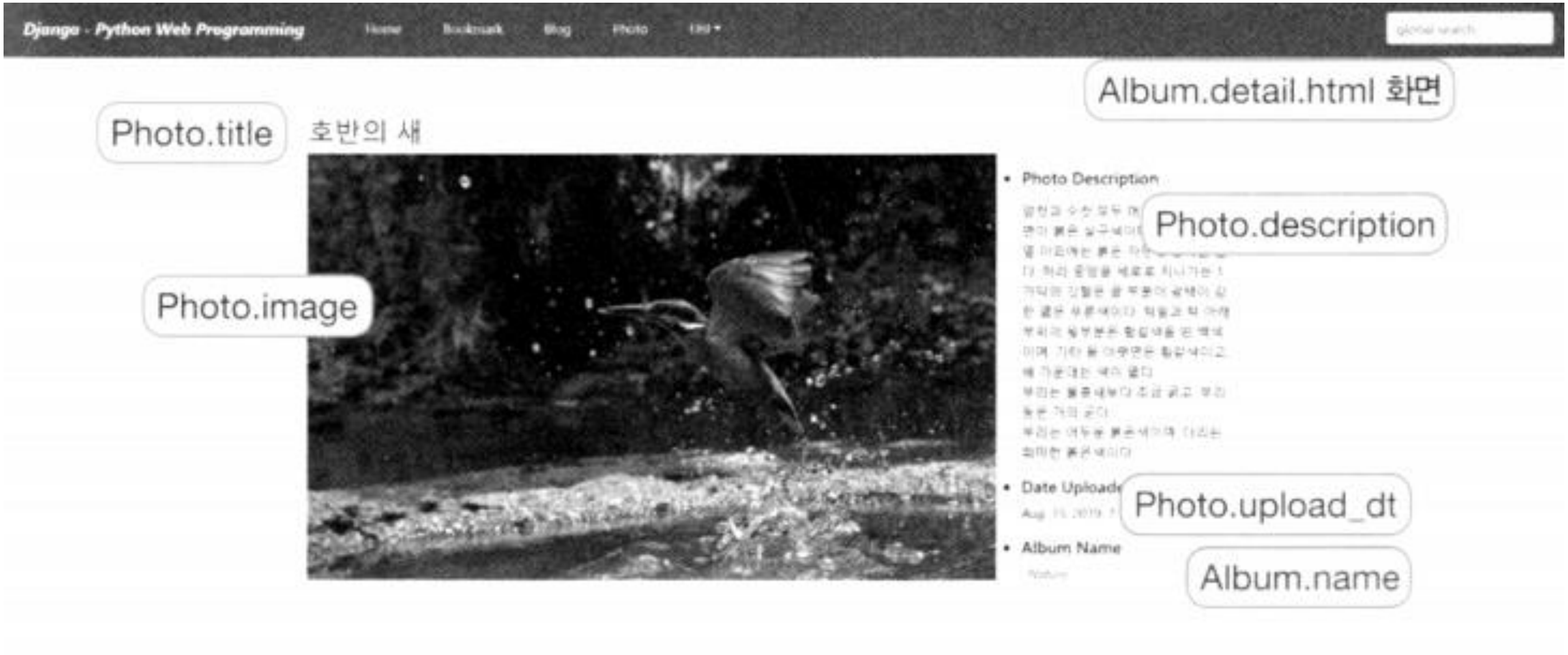
애플리케이션 설계하기

❖ 화면 설계



애플리케이션 설계하기

❖ 화면 설계



Album.detail.html 화면

Photo.title
호반의 새

Photo.image

- Photo Description

[illegible]

Photo.description

- Date Uploaded

Aug. 15, 2019, [illegible]

Photo.upload_dt

- Album Name

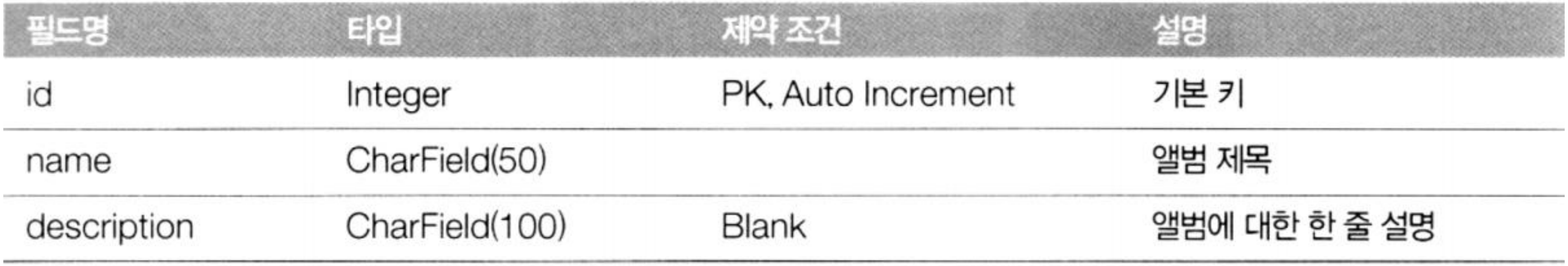
Nature

Album.name

애플리케이션 설계하기

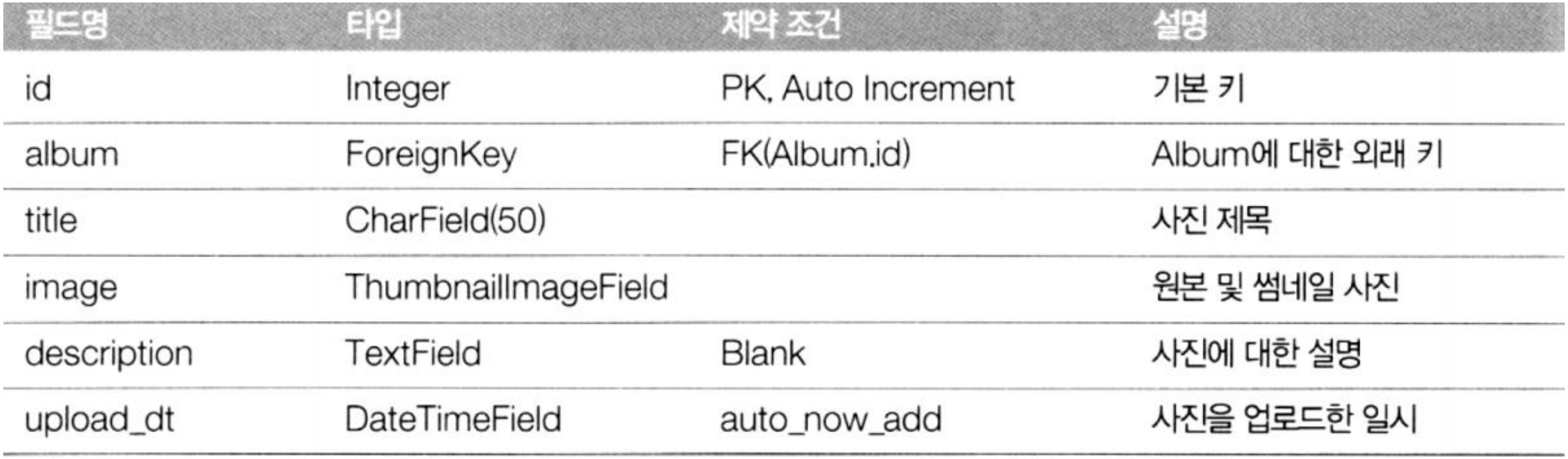
❖ 테이블 설계

- Album 모델 클래스



필드명	타입	제약 조건	설명
id	Integer	PK, Auto Increment	기본 키
name	CharField(50)		앨범 제목
description	CharField(100)	Blank	앨범에 대한 한 줄 설명

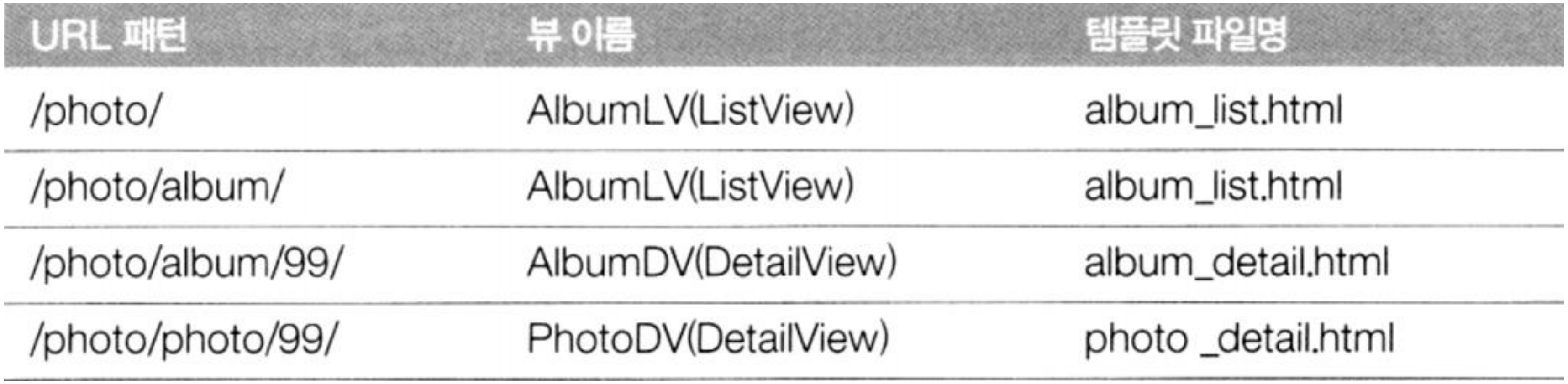
- Photo 모델 클래스



필드명	타입	제약 조건	설명
id	Integer	PK, Auto Increment	기본 키
album	ForeignKey	FK(Album.id)	Album에 대한 외래 키
title	CharField(50)		사진 제목
image	ThumbnailImageField		원본 및 썸네일 사진
description	TextField	Blank	사진에 대한 설명
upload_dt	DateTimeField	auto_now_add	사진을 업로드한 일시

애플리케이션 설계하기

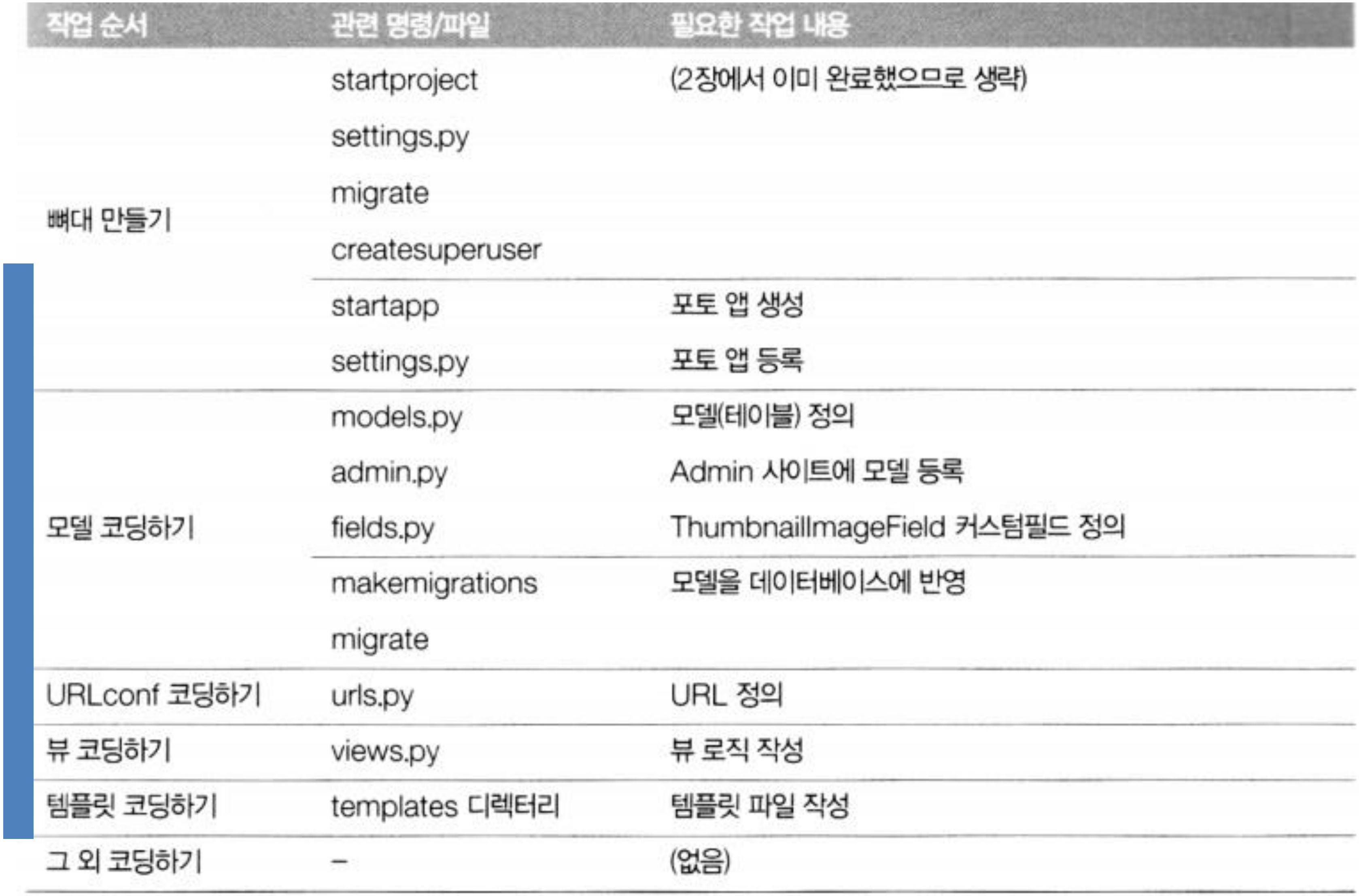
❖ URL 설계



URL 패턴	뷰 이름	템플릿 파일명
/photo/	AlbumLV(ListView)	album_list.html
/photo/album/	AlbumLV(ListView)	album_list.html
/photo/album/99/	AlbumDV(DetailView)	album_detail.html
/photo/photo/99/	PhotoDV(DetailView)	photo _detail.html

애플리케이션 설계하기

❖ 작업 절차



작업 순서	관련 명령/파일	필요한 작업 내용
뼈대 만들기	startproject	(2장에서 이미 완료했으므로 생략)
	settings.py	
	migrate	
	createsuperuser	
	startapp	포토 앱 생성
	settings.py	포토 앱 등록
모델 코딩하기	models.py	모델(테이블) 정의
	admin.py	Admin 사이트에 모델 등록
	fields.py	ThumbnailImageField 커스텀필드 정의
	makemigrations	모델을 데이터베이스에 반영
	migrate	
URLconf 코딩하기	urls.py	URL 정의
뷰 코딩하기	views.py	뷰 로직 작성
템플릿 코딩하기	templates 디렉터리	템플릿 파일 작성
그 외 코딩하기	–	(없음)

개발 코딩하기

❖ 앱 생성

```
$ python manage.py startapp photo
```

❖ mysite/settings.py : 앱 등록

```
INSTALLED_APPS = [
   :
   'photo.apps.PhotoConfig',
]

:

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

개발 코딩하기

❖ photo/models.py : 모델 정의

```
from django.db import models
from django.urls import reverse

from photo.fields import ThumbnailImageField


class Album(models.Model):
  name = models.CharField('NAME', max_length=30)
  description = models.CharField('One Line Description', max_length=100,
                                 blank=True)

  class Meta:
    ordering = ('name',)

  def __str__(self):
    return self.name

  def get_absolute_url(self):
    return reverse('photo:album_detail', args=(self.id,))
```

개발 코딩하기

❖ photo/models.py : 모델 정의

```
class Photo(models.Model):
  album = models.ForeignKey(Album, on_delete=models.CASCADE)
  title = models.CharField('TITLE', max_length=30)
  description = models.TextField('Photo Description', blank=True)
  image = ThumbnailImageField('IMAGE', upload_to='photo/%Y/%m')
  upload_dt = models.DateTimeField('UPLOAD DATE', auto_now_add=True)

  class Meta:
    ordering = ('title',)

  def __str__(self):
    return self.title

  def get_absolute_url(self):
    return reverse('photo:photo_detail', args=(self.id,))
```

개발 코딩하기

❖ photo/models.py : 모델 정의

```
class Photo(models.Model):
  album = models.ForeignKey(Album, on_delete=models.CASCADE)
  title = models.CharField('TITLE', max_length=30)
  description = models.TextField('Photo Description', blank=True)
  image = ThumbnailImageField('IMAGE', upload_to='photo/%Y/%m')
  upload_dt = models.DateTimeField('UPLOAD DATE', auto_now_add=True)

  class Meta:
    ordering = ('title',)

  def __str__(self):
    return self.title

  def get_absolute_url(self):
    return reverse('photo:photo_detail', args=(self.id,))
```

개발 코딩하기

❖ photo/admin.py

```
from django.contrib import admin
from photo.models import Album, Photo

class PhotoInline(admin.StackedInline):
  model = Photo
  extra = 2

@admin.register(Album)
class AlbumAdmin(admin.ModelAdmin):
  inlines = (PhotoInline,)
  list_display = ('id', 'name', 'description')


@admin.register(Photo)
class PhotoAdmin(admin.ModelAdmin):
  list_display = ('id', 'title', 'upload_dt')
```

개발 코딩하기

❖ photo/field.py : 사용자 정의 필드

```
import os
from PIL import Image
from django.db.models import ImageField
from django.db.models.fields.files import ImageFieldFile


class ThumbnailImageFieldFile(ImageFieldFile):
  def _add_thumb(self, s):
    parts = s.split('.')
    parts.insert(-1, 'thumb')
    if parts[-1].lower() not in ('jpeg', 'jpg'):
      parts[-1] = 'jpg'
    return '.'.join(parts)

  @property
  def thumb_path(self):
    return self._add_thumb(self.path)

  @property
  def thumb_url(self):
    return self._add_thumb(self.url)
```

개발 코딩하기

❖ photo/field.py : 사용자 정의 필드

```
def save(self, name, content, save=True):
  super().save(name, content, save)

  img = Image.open(self.path)
  size = (self.field.thumb_width, self.field.thumb_height)
  img.thumbnail(size)
  background = Image.new('RGB', size, (255, 255, 255))
  box = (int((size[0]-img.size[0])/2), int((size[1]-img.size[1])/2))
  background.paste(img, box)
  background.save(self.thumb_path, 'JPEG')

def delete(self, save=True):
  if os.path.exists(self.thumb_path):
    os.remove(self.thumb_path)
  super().delete(save)
```

개발 코딩하기

❖ photo/field.py : 사용자 정의 필드

```
class ThumbnailImageField(ImageField):
  attr_class = ThumbnailImageFieldFile

  def __init__(self, verbose_name=None, thumb_width=128, thumb_height=128,
               **kwargs):
    self.thumb_width, self.thumb_height = thumb_width, thumb_height

    super().__init__(verbose_name, **kwargs)
```

개발 코딩하기

❖ 모델을 데이터베이스에 반영

```
$ python manage.py makemigraions photo
$ python manage.py migrate
```

개발 코딩하기

❖ photo/views.py

```
from django.views.generic import ListView, DetailView
from photo.models import Album, Photo

class AlbumLV(ListView):
    model = Album

class AlbumDV(DetailView):
    model = Album

class PhotoDV(DetailView):
    model = Photo
```

개발 코딩하기

❖ photo/urls.py

```
from django.urls import path

from photo import views

app_name = 'photo'
urlpatterns = [

  # Example: /photo/
  path('', views.AlbumLV.as_view(), name='index'),

  # Example: /photo/album/, same as /photo/
  path('album', views.AlbumLV.as_view(), name='album_list'),

  # Example: /photo/album/99/
  path('album/<int:pk>/', views.AlbumDV.as_view(), name='album_detail'),

  # Example: /photo/photo/99/
  path('photo/<int:pk>/', views.PhotoDV.as_view(), name='photo_detail'),

]
```

개발 코딩하기

❖ mysite/urls.py

```
from django.conf import settings
from django.conf.urls.static import static
from django.contrib import admin
from django.urls import path, include
from mysite.views import HomeView

urlpatterns = [
  path('admin/', admin.site.urls),

  path('', HomeView.as_view(), name='home'),
  path('bookmark/', include('bookmark.urls')),
  path('blog/', include('blog.urls')),
  path('photo/', include('photo.urls')),

] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

개발 코딩하기

❖ photo/templates/photo/album_list.html

```
{% extends "base.html" %}

{% block title %}album_list.html{% endblock %}

{% block extra-style %}
<style>
.thumbnail {
  border: 3px solid #ccc;
}
</style>
{% endblock %}

{% block content %}

  {% for item in object_list %}

  <div class="mt-5">
    <a class="h2" href="{% url 'photo:album_detail' item.id %}">
      {{ item.name }}</a>&emsp;
    <span class="font-italic h5">{{ item.description }}</span>
  </div>
```

개발 코딩하기

❖ photo/templates/photo/album_list.html

```
  <hr style="margin: 0 0 20px 0;">

  <div class="row">
    {% for photo in item.photo_set.all|slice:":5" %}
    <div class="ml-5">
      <div class="thumbnail">
        <a href="{{ photo.get_absolute_url }}">
          <img src="{{ photo.image.thumb_url }}" style="width: 100%;">
        </a>
      </div>
    </div>
    {% endfor %}
  </div>

  {% endfor %}

{% endblock %}
```

개발 코딩하기

❖ photo/templates/photo/album_detail.html

```
{% extends "base.html" %}

{% block title %}album_detail.html{% endblock %}

{% block extra-style %}
<style>
.thumbnail {
  border: 5px solid #ccc;
}
</style>
{% endblock %}

{% block content %}

  <div class="mt-5">
    <span class="h2">{{ object.name }}&emsp;</span>
    <span class="h5 font-italic">{{ object.description }}</span>
  </div>

  <hr style="margin: 0 0 20px 0;">
```

개발 코딩하기

❖ photo/templates/photo/album_detail.html

```
  <div class="row">

    {% for photo in object.photo_set.all %}
    <div class="col-md-3 mb-5">
      <div class="thumbnail">
        <a href="{{ photo.get_absolute_url }}">
          <img src="{{ photo.image.thumb_url }}" style="width: 100%;">
        </a>
      </div>
      <ul>
        <li class="font-italic">{{ photo.title }}</li>
        <li class="font-italic">{{ photo.upload_dt|date:"Y-m-d" }}</li>
      </ul>
    </div>
    {% endfor %}

  </div>

{% endblock %}
```

개발 코딩하기

❖ photo/templates/photo/photo_detail.html

```
{% extends "base.html" %}

{% block title %}photo_detail.html{% endblock %}

{% block content %}

  <h2 class="mt-5">{{ object.title }}</h2>

  <div class="row">
    <div class="col-md-9">
      <a href="{{ object.image.url }}">
        <img src="{{ object.image.url }}" style="width: 100%;">
      </a>
    </div>
```

개발 코딩하기

❖ **photo/templates/photo/photo_detail.html**

```
    <ul class="col-md-3 mt-3">
      <li class="h5">Photo Description</li>
        {% if object.description %}<p>{{ object.description|linebreaks }}</p>
        {% else %}<p>(blank)</p>{% endif %}
      <li class="h5">Date Uploaded</li>
        <p class="font-italic">{{ object.upload_dt }}</p>
      <li class="h5">Album Name</li>
        <p class="font-italic">
           <a href="{% url 'photo:album_detail' object.album.id %}">
              {{ object.album.name }}</a>
        </p>
    </ul>
  </div>

{% endblock %}
```

개발 코딩하기

❖ menu.html

```
:

 <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item mx-1 btn btn-primary">
        <a class="nav-link text-white" href="{% url 'home' %}">Home</a></li>
      <li class="nav-item mx-1 btn btn-primary">
        <a class="nav-link text-white"
           href="{% url 'bookmark:index' %}">Bookmark</a></li>
      <li class="nav-item mx-1 btn btn-primary">
        <a class="nav-link text-white"
           href="{% url 'blog:index' %}">Blog</a></li>
      <li class="nav-item mx-1 btn btn-primary">
        <a class="nav-link text-white"
            href="{% url 'photo:index' %}">Photo</a></li>
```