실습 예제 확장하기

새로운 애플리케이션 만들기

❖ 뼈대 만들기

- books 앱 만들기

```
> python manage.py startapp books
```

- mysite/settings.py

```
INSTALLED_APPS = [
    :
    'books'
]
```

새로운 애플리케이션 만들기

❖ 애플리케이션 설계하기

- 화면 설계
  - index.html
  - book_list.html
  - book_detail.html

새로운 애플리케이션 만들기

❖ 애플리케이션 설계하기

- Book 테이블

id	integer NotNull, PK, AutoIncrement	Primary Key
title	varchar(100) NotNull	책 제목
authors	integer NotNull, FK(Author.id), index	Many-to-Many
publisher	integer NotNull, FK(Publisher.id), index	Foreign Key
publication_date	date NotNull	책 출판일

- Author 테이블

id	integer	NotNull, PK, AutoIncrement	Primary Key
salutation	varchar(100)	NotNull	저자 인사말
name	varchar(50)	NotNull	저자 이름
email	email	NotNull	저자 이메일

- Publisher 테이블

id	integer	NotNull, PK, AutoIncrement	Primary Key
name	varchar(50)	NotNull	출판사 이름
address	varchar(200)	NotNull	출판사 주소
website	url	NotNull	출판사 홈페이지

새로운 애플리케이션 만들기

❖ 애플리케이션 설계하기

- URL 설계
  - /books/ BooksModelView index.html
  - /books/book/ BookList book_list.html
  - /books/book/3/ BookDetail book_detail.html

새로운 애플리케이션 만들기

❖ Model 코딩하기

- books/models.py

```
from django.db import models

class Book(models.Model):
    title              = models.CharField(max_length=100)
    authors            = models.ManyToManyField('Author')
    publisher          = models.ForeignKey('Publisher')
    publication_date   = models.DateField()

    def __str__(self):
        return self.title

    def __repr__(self):
        return self.title
```

새로운 애플리케이션 만들기

❖ Model 코딩하기

- books/models.py

```
class Author(models.Model):
    salutation  = models.CharField(max_length=100)
    name        = models.CharField(max_length=50)
    email       = models.EmailField()

    def __str__(self):
        return self.name

    def __repr__(self):
        return self.name
```

새로운 애플리케이션 만들기

❖ Model 코딩하기

- books/models.py

```
class Publisher(models.Model):
    name     = models.CharField(max_length=50)
    address  = models.CharField(max_length=100)
    website  = models.URLField()

    def __str__(self):
        return self.name

    def __repr__(self):
        return self.name
```

새로운 애플리케이션 만들기

❖ Model 코딩하기

- books/admin.py

```
from django.contrib import admin
from books.models import Book, Author, Publisher

admin.site.register(Book)
admin.site.register(Author)
admin.site.register(Publisher)
```

- 데이터베이스 반영

```
> python manage.py makemigrations books
> python migrate
```

새로운 애플리케이션 만들기

❖ 테스트 데이터 구축/확인

- Author 테이블
  - Kim Seok Hun/I'm a python programmer/shkshya@daum.net
  - Eric Gamma/Welcome to Gang of Four/ericgamma@gmail.com
  - Johsua Bloch/Java Great Programer/joshua@gmail.com

- Publisher 테이블
  - Hanbit Media. Inc/Seoul. Korea/http://www.hanb.co.kr/
  - O'Reilly/Sanfrancisco. US/http://www.oreilly.com/
  - Pearson Education Inc./United States/http://pearson.com/

- Book 테이블
  - Python Web Programming/Kim Seok Hun/Hanbit Media. Inc/2019-03-01
  - Design Patterns/Eric Gamma/O'Reilly/2005-12-25
  - Effective Java/Joshua Bloch/Pearson Education Inc./2008-09-02

새로운 애플리케이션 만들기

❖ URLconf 구성하기

- books/urls.py

```
from django.urls import path
from . import views

app_name = 'books'

urlpatterns = [
    # /books/
    path('', views.BooksModelView.as_view(), name='index'),

    # /books/book/
    path('book/', views.BookList.as_view(), name='book_list'),
    # /books/book/99/
    path('book/<int:pk>/', views.BookDetail.as_view(), name='book_detail'),

    # /books/author/
    path('author/', views.AuthorList.as_view(), name='author_list'),
    # /books/author/99/
    path('author/<int:pk>/', views.AuthorDetail.as_view(),
         name='author_detail'),
```

새로운 애플리케이션 만들기

❖ URLconf 구성하기

- books/urls.py

```
    # /books/publisher/
    path('publisher/', views.PublisherList.as_view(), name='publisher_list'),
    # /books/publisher/99/
    path('publisher/<int:pk>/', views.PublisherDetail.as_view(),
         name='publisher_detail'),
]
```

새로운 애플리케이션 만들기

❖ URLconf 구성하기

- mysite/urls.py

```
from django.contrib import admin
from django.urls import path, include
from mysite import views


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

    path('', views.HomeView.as_view(), name='home'),
    path('polls/', include('polls.urls')),
    path('books/', include('books.urls')),
]
```

새로운 애플리케이션 만들기

❖ Template 코딩하기

- 부모 템플릿
  - templates/
    - base.html
    - base_books.html

  - 블럭 정의
    - title
    - sidebar
    - content
  - mysite/settings.py에 해당 경로 등록

- 자식 템플릿
  - 부모를 상속받아 정의
    - {% extends "base.html" %}
  - books/templates/books에 배치

새로운 애플리케이션 만들기

❖ Template 코딩하기

- mysite/settings.py

```
import os
:

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

새로운 애플리케이션 만들기

❖ Template 코딩하기

- templates/base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    {% load static %}
    <link rel="stylesheet" href="{% static 'admin/css/base.css' %}" />
    <title>{% block title %}My Amazing Site{% endblock %}</title>
</head>
<body>
    <div id="sidebar">
        {% block sidebar %}
        <ul>
            <li><a href="/">Project_Home</a></li>
            <li><a href="/admin/">Admin</a></li>
        </ul>
        {% endblock %}
        <br>
    </div>

    <div id="content">
        {% block content %}{% endblock %}
    </div>
</body>
</html>
```

새로운 애플리케이션 만들기

❖ Template 코딩하기

- templates/base_books.html

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

{% block title %}Books Application Site{% endblock %}

{% block sidebar %}
{{ block.super }}
<ul>
    <li><a href="/books/">Books_Home</a></li>
</ul>
{% endblock %}
```

새로운 애플리케이션 만들기

❖ Template 코딩하기

- books/templates/books/index.html

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

{% block content %}
    <h2>Books Management System</h2>
    <ul>
        {% for modelname in model_list %}
          {% with "books:"|add:modelname|lower|add:"_list" as urlvar %}
            <li><a href="{% url urlvar %}">{{ modelname }}</a></li>
          {% endwith %}
        {% endfor %}
    </ul>
{% endblock content %}
```

새로운 애플리케이션 만들기

❖ Template 코딩하기

- books/templates/books/book_list.html

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

{% block content %}
    <h2>Book List</h2>
    <ul>
        {% for book in object_list %}
            <li><a href="{% url 'books:book_detail' book.id %}">
                {{ book.title }}</a></li>
        {% endfor %}
    </ul>
{% endblock content %}
```

새로운 애플리케이션 만들기

❖ Template 코딩하기

- books/templates/books/book_detail.html

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

{% block content %}

<h1>{{ object.title }}</h1>
<br>
<li>Authors:
{% for author in object.authors.all %}
  {{ author }}
  {% if not forloop.last %},{% else %}{% endif %}
{% endfor %}
</li>

<li>Publisher: {{ object.publisher }}</li>
<li>Publication date: {{ object.publication_date }}</li>

{% endblock content %}
```

새로운 애플리케이션 만들기

❖ Template 코딩하기

- books/templates/books/publisher_list.html

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

{% block content %}
    <h2>Publisher List</h2>
    <ul>
        {% for publisher in object_list %}
            <li><a href="{% url 'books:publisher_detail' publisher.id %}">
                  {{ publisher.name }}</a></li>
        {% endfor %}
    </ul>
{% endblock content %}
```

새로운 애플리케이션 만들기

❖ Template 코딩하기

- books/templates/books/publisher_detail.html

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

{% block content %}

<h1>{{ object.name }}</h1>
<p>{{ object.website }}</p>
<li>Address: {{ object.address }}</li>

{% endblock content %}
```

새로운 애플리케이션 만들기

❖ Template 코딩하기

- books/templates/books/author_list.html

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

{% block content %}
    <h2>Author List</h2>
    <ul>
        {% for author in object_list %}
            <li><a href="{% url 'books:author_detail'
author.id %}">{{ author.name }}</a></li>
        {% endfor %}
    </ul>
{% endblock content %}
```

새로운 애플리케이션 만들기

❖ Template 코딩하기

- books/templates/books/author_detail.html

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

{% block content %}

<h1>{{ object.name }}</h1>
<p>{{ object.salutation }}</p>
<li>Email: {{ object.email }}</li>

{% endblock content %}
```

새로운 애플리케이션 만들기

❖ 클래스형 View 코딩하기

- books/views.py

```
from django.views.generic.base import TemplateView
from django.views.generic import ListView
from django.views.generic import DetailView
from books.models import Book, Author, Publisher


#--- TemplateView
class BooksModelView(TemplateView):
    template_name = 'books/index.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['model_list'] = ['Book', 'Author', 'Publisher']
        return context
```

새로운 애플리케이션 만들기

❖ 클래스형 View 코딩하기

- books/views.py

```
#--- ListView
class BookList(ListView):
    model = Book


class AuthorList(ListView):
    model = Author


class PublisherList(ListView):
    model = Publisher
```

새로운 애플리케이션 만들기

❖ 클래스형 View 코딩하기

- books/views.py

```
#--- DetailView
class BookDetail(DetailView):
    model = Book


class AuthorDetail(DetailView):
    model = Author


class PublisherDetail(DetailView):
    model = Publisher
```

프로젝트 첫 페이지 만들기

❖ 프로젝트 첫 페이지 설계

프로젝트 첫 페이지 만들기

❖ URLconf 코딩하기

- mysite/urls.py

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

    path('', views.HomeView.as_view(), name='home'),
    path('polls/', include('polls.urls')),
    path('books/', include('books.urls')),
]
```

프로젝트 첫 페이지 만들기

❖ View 코딩하기

- mysite/views.py

```
from django.views.generic.base import TemplateView

#--- TemplateView
class HomeView(TemplateView):

    template_name = 'home.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['app_list'] = ['polls', 'books']
        return context
```

프로젝트 첫 페이지 만들기

❖ Template 코딩하기

- templates/home.html

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

{% block content %}
    <h2>Django Applications</h2>
    <ul>
      {% for appname in app_list %}
        {% with appname|add:":"|add:"index" as urlvar %}
            <li><a href="{% url urlvar %}">{{ appname }}</a></li>
        {% endwith %}
      {% endfor %}
    </ul>
{% endblock content %}
```

polls 애플리케이션 - 클래스형 뷰로 변경하기

❖ URLconf 코딩하기

- polls/urls.py

```
from django.urls import path
from polls import views

app_name = 'polls'
urlpatterns = [
    # /polls/
    path('', views.IndexView.as_view(), name='index'),

    # /polls/99/
    path('<int:pk>/', views.DetailView.as_view(), name='detail'),

    # /polls/99/results/
    path('<int:pk>/results/', views.ResultsView.as_view(), name='results'),

    # /polls/99/vote/
    path('<int:question_id>/vote/', views.vote, name='vote'),
]
```

polls 애플리케이션 - 클래스형 뷰로 변경하기

❖ View 코딩하기

- polls/views.py

```
from django.shortcuts import get_object_or_404, render
from django.http import HttpResponseRedirect
from django.urls import reverse
from django.views import generic

from polls.models import Choice, Question

import logging
logger = logging.getLogger(__name__)


#--- Class-based View
class IndexView(generic.ListView):
    template_name = 'polls/index.html'
    context_object_name = 'latest_question_list'

    def get_queryset(self):
        """Return the last five published questions."""
        return Question.objects.order_by('-pub_date')[:5]
```

polls 애플리케이션 - 클래스형 뷰로 변경하기

❖ View 코딩하기

- polls/views.py

```
class DetailView(generic.DetailView):
    model = Question
    template_name = 'polls/detail.html'


class ResultsView(generic.DetailView):
    model = Question
    template_name = 'polls/results.html'
```

polls 애플리케이션 - 클래스형 뷰로 변경하기

❖ View 코딩하기

- polls/views.py

```
#--- Funtion-based View
def vote(request, question_id):
    logger.debug("vote().question_id: %s" % question_id)
    question = get_object_or_404(Question, pk=question_id)

    try:
        selected_choice = question.choice_set.get(pk=request.POST['choice'])
    except (KeyError, Choice.DoesNotExist):
        return render(request, 'polls/detail.html', {
            'question': question,
            'error_message': "You didn't select a choice.",
        })
    else:
        selected_choice.votes += 1
        selected_choice.save()
        return HttpResponseRedirect(reverse('polls:results',
                                    args=(question.id,)))
```

polls 애플리케이션 - 클래스형 뷰로 변경하기

❖ 템플릿 코딩하기

- templates/base_polls.html

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

{% block title %}Polls Application Site{% endblock %}

{% block sidebar %}
{{ block.super }}
<ul>
    <li><a href="/polls/">Polls_Home</a></li>
</ul>
{% endblock %}
```

polls 애플리케이션 - 클래스형 뷰로 변경하기

❖ 템플릿 코딩하기

- polls/templates/polls/index.html

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

{% block content %}

<h2>Polls Question List</h2>

{% if latest_question_list %}
    <ul>
    {% for question in latest_question_list %}
        <li><a href="{% url 'polls:detail'
question.id %}">{{ question.question_text }}</a></li>
    {% endfor %}
    </ul>
{% else %}
    <p>No polls are available.</p>
{% endif %}

{% endblock %}
```

polls 애플리케이션 - 클래스형 뷰로 변경하기

❖ 템플릿 코딩하기

- polls/templates/polls/detail.html

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

{% block content %}

<h1>{{ question.question_text }}</h1>

{% if error_message %}<p><strong>{{ error_message }}</strong></p>{% endif %}

<form action="{% url 'polls:vote' question.id %}" method="post">
{% csrf_token %}
{% for choice in question.choice_set.all %}
    <input type="radio" name="choice" id="choice{{ forloop.counter }}"
           value="{{ choice.id }}" />
    <label for="choice{{ forloop.counter }}">
       {{ choice.choice_text }}</label><br />
{% endfor %}
<input type="submit" value="Vote" />
</form>

{% endblock %}
```

polls 애플리케이션 - 클래스형 뷰로 변경하기

❖ 템플릿 코딩하기

- polls/templates/polls/results.html

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

{% block content %}

<h1>{{ question.question_text }}</h1>

<ul>
{% for choice in question.choice_set.all %}
    <li>{{ choice.choice_text }} -- {{ choice.votes }}
        vote{{ choice.votes|pluralize }}</li>
{% endfor %}
</ul>

<a href="{% url 'polls:detail' question.id %}">Vote again?</a>

{% endblock %}
```