Django 웹 프레임워크

애플리케이션 개발하기 – Model 코딩

Model 코딩

- 모델 작업은 데이터베이스에 테이블을 생성하는 작업
- >notepad models.py // 테이블을 정의함
- >notepad admins.py // 정의된 테이블이 Admin 화면에 보이게 함
- >python manage.py makemigrations // 데이터베이스에 변경이 필요한 사항을 추출함
- >python manage.py migrate // 데이터베이스에 변경사항을 반영함
- >python manage.py runserver // 현재까지 작업을 개발용 웹 서버로 확인함

애플리케이션 개발하기 – Model 코딩

테이블 정의

- polls 애플리케이션은 Question과 Choice 두 개의 테이블이 필요
- 테이블은 **models.py** 파일에 정의

테이블 컬럼명	컬럼 타입	장고의 클래스 변수	장고의 필드 클래스
id	integer	(id)	(PK는 장고에서 자동 생성해줌)
question_text	varchar(200)	question_text	models.CharField(max_length=200)
pub_date	datetime	pub_date	models.DateTimeField('date published')

컬럼명	타입	장고의 클래스 변수	장고의 필드 클래스
id	integer	(id)	(PK는 장고에서 자동 생성해줌)
choice_text	varchar(200)	choice_text	models.CharField(max_length=200)
votes	integer	votes	models.IntegerField(default=0)
question_id	integer	question	models.ForeighKey(Question)

애플리케이션 개발하기 – Model 코딩

테이블 정의

- 테이블은 **models.py** 파일에 정의

```
from django.db import models

class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

    def __str__(self):
        return self.question_text

class Choice(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)

    def __str__(self):
        return self.choice_text
```

애플리케이션 개발하기 – Model 코딩

Admin 사이트에 테이블 반영

- models.py 파일에서 정의한 테이블도 admin 사이트에 보이도록 등록
- admin.py 파일에 등록

```
from django.contrib import admin
from polls.models import Question, Choice



admin.site.register(Question)
admin.site.register(Choice)
```

  - models.py 모듈에서 정의한 Question, Choice 클래스를 임포트 하 고 ,

    admin.site.register() 함수를 사용하여 임포트 한 클래스를 Admin 사이트에 등록

애플리케이션 개발하기 – Model 코딩

데이터베이스 변경사항 반영

- 테이블의 신규 생성, 테이블의 정의 변경 등 데이터베이스에 변경이 필요한 사항이 있으면, 이를 데이터베이스에 실제로 반영해주는 작업
- 변경사항을 데이터베이스에 반영
- (base) C:₩MyTest₩projectsite>python manage.py makemigrations

```
(base) C:₩MyTest₩projectsite>python manage.py makemigrations
Migrations for 'polls':
  polls₩migrations₩0001_initial.py
    - Create model Question
    - Create model Choice
```

애플리케이션 개발하기 – Model 코딩

데이터베이스 변경사항 반영

- makemigrations 명령에 의해 polls/migrations 디렉토리 하위에 마이그레이션 파일들이 생김

```
(base) C:₩MyTest₩projectsite₩polls₩migrations>dir
 C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: 421A-9EBA

 C:₩MyTest₩projectsite₩polls₩migrations 디렉터리

2021-08-24  오후 06:58    <DIR>          .
2021-08-24  오후 06:58    <DIR>          ..
2021-08-24  오후 06:58             1,113 0001_initial.py
2021-08-23  오후 10:41                 0 __init__.py
2021-08-23  오후 11:30    <DIR>          __pycache__
               2개 파일               1,113 바이트
               3개 디렉터리  20,635,996,160 바이트 남음
```

애플리케이션 개발하기 – Model 코딩

데이터베이스 변경사항 반영

- 마이그레이션 파일들을 이용해 migrate 명령으로 데이터베이스에 테이블을 생성

- (base) C:₩MyTest₩projectsite>python manage.py migrate

```
(base) C:₩MyTest₩projectsite>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, polls, sessions
Running migrations:
  Applying polls.0001_initial... OK
```

애플리케이션 개발하기 – Model 코딩

작업 확인

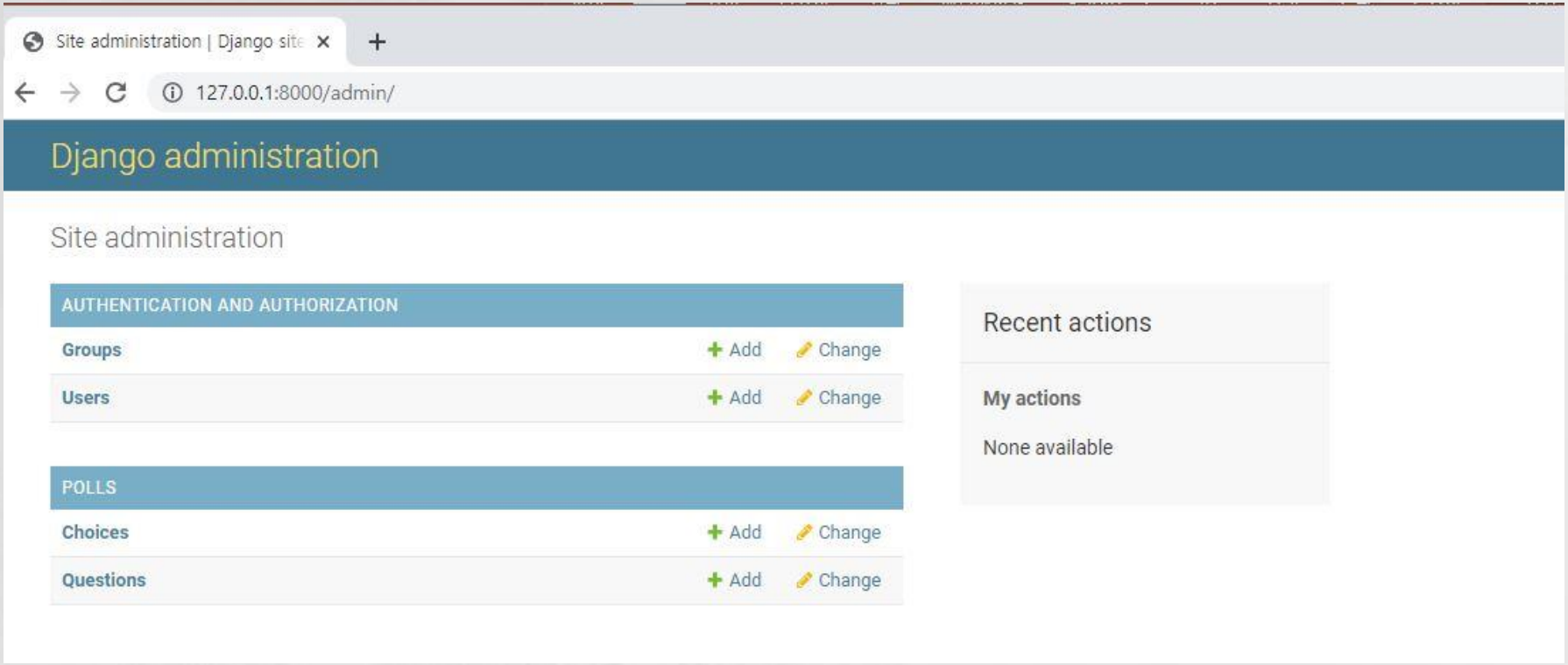
- 정상적으로 잘 처리되었는지 확인하기 위해 Admin 사이트에 접속
- 하나의 창에서 작업해도 되지만 runserver 용으로 별도의 cmd창을 열어 사용하는 것이 편리
  - (base) C:₩MyTest₩projectsite>python manage.py runserver 127.0.0.1:8000
  - (base) C:₩MyTest₩projectsite>python manage.py runserver
  - (base) C:₩MyTest₩projectsite>python manage.py runserver 0:8000

```
(base) C:₩MyTest>cd projectsite

(base) C:₩MyTest₩projectsite>python manage.py runserver 127.0.0.1:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 23, 2021 - 23:40:17
Django version 3.2.6, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

애플리케이션 개발하기 – Model 코딩

작업 확인

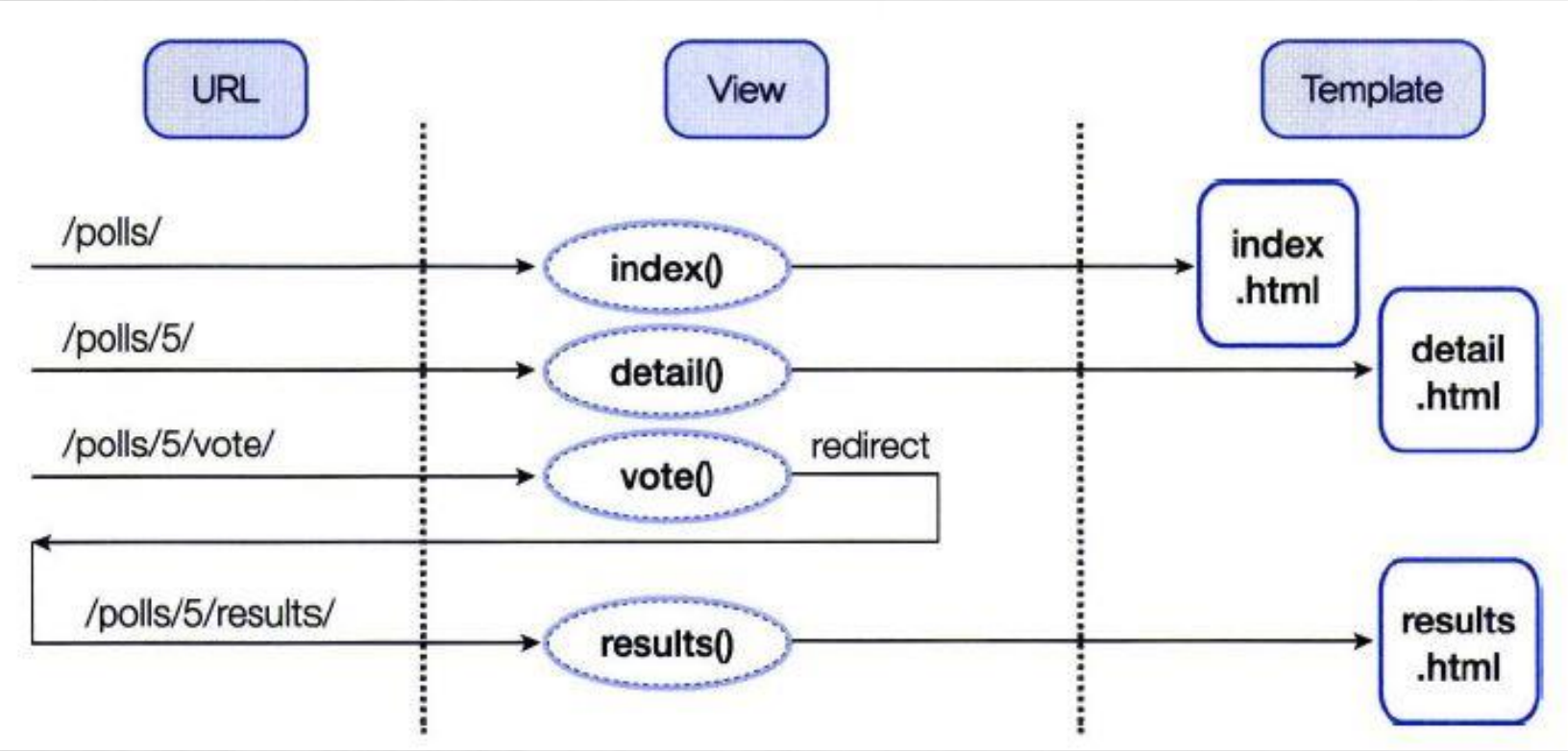
- Questions와 Choices 테이블을 만든 모델이 정상적으로 등록



애플리케이션 개발하기 –View 및 Template

처리 흐름 설계

- 사용자에게 보여지는 페이지가 3개이므로, 3개의 템플릿 파일이 필요



애플리케이션 개발하기 –View 및 Template

처리 흐름 설계

- URL/뷰 매핑을 URLconf 라고 하며 urls.py 파일에 작성

URL 패턴	뷰 이름	뷰가 처리하는 내용
/polls/	index( )	index.html 템플릿을 보여줍니다.
/polls/5/	detail( )	detail.html 템플릿을 보여줍니다.
/polls/5/vote/	vote( )	detail.html에 있는 폼을 POST 방식으로 처리합니다.
/polls/5/results/	results( )	results.html 템플릿을 보여줍니다.
/admin/	(장고 기능)	Admin 사이트를 보여줍니다(장고에서 기본으로 제공함).

애플리케이션 개발하기 –View 및 Template

처리 흐름 설계

- 로직의 흐름상 URLconf를 먼저 코딩한 후에 뷰,템플릿 또는 템플릿, 뷰 순서로 코딩하는 것이 일반적
- urls.py 작성 // URLconf 내용을 코딩
- views.index() 함수 작성 // index.html 템플릿도 같이 작성
- view s.detail() 함수 작성 // d etail.htm l 템플릿도 같이 작성
- views.voteO 함수 작성 // 리다이렉션 처리 들어있음
- view s.results0 함수 작성 // results.htm l 템플릿도 같이 작성

애플리케이션 개발하기 –View 및 Template

URLconf 코딩

- URLconf 설계 내용에 따르면, Admin사이트까지 포함해서 5개의 URL과 뷰가 필요

- urls.py(C:₩MyTest₩projectsite₩mysite) 파일에 코딩

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

    # 추가
    path('polls/', include('polls.urls')),
]
```

애플리케이션 개발하기 –View 및 Template

URLconf 코딩

- urls.py(C:₩MyTest₩projectsite₩mysite₩polls) 파일에 코딩

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

app_name = 'polls'
urlpatterns = [
    path('', views.index, name='index'),       # /polls/
    path('<int:question_id>/', views.detail, name='detail'),       # /polls/5/
    path('<int:question_id>/results/', views.results, name='results'),     # /polls/5/results/
    path('<int:question_id>/vote/', views.vote, name='vote'),      # /polls/5/vote/
]
```

애플리케이션 개발하기 –View 및 Template

뷰 함수 index() 및 템플릿 작성

- 뷰함수와 템플릿은 서로에게 영향을 미치기 때문에 보통 같이 작업
- 내용을 구현하기 위해 템플릿 파일 index.html

애플리케이션 개발하기 –View 및 Template

뷰 함수 index() 및 템플릿 작성

- 내용을 구현하기 위해 템플릿 파일 index.html

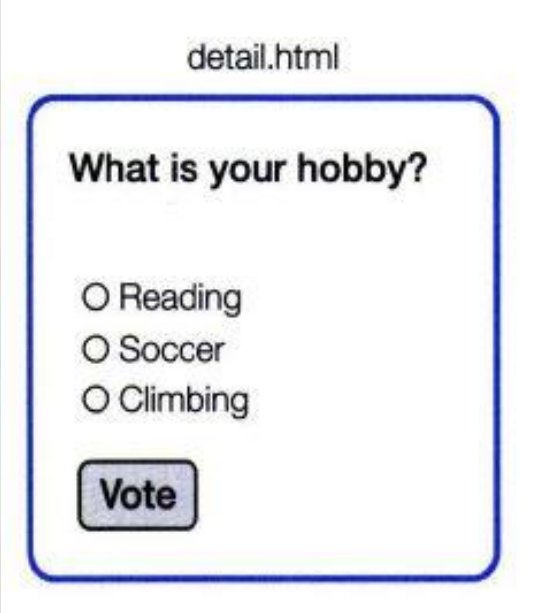

```
{% if latest_question_list %}
    <ul>
    {% for question in latest_question_list %}
        <li><a href="/polls/{{ question.id }}/">{{ question.question_text }}</a></li>
    {% endfor %}
    </ul>
{% else %}
    <p>No polls are available.</p>
{% endif %}
```

```
<ul>
    <li><a href="/polls/1/">What is your hobby?</a></li>
    <li><a href="/polls/2/">Who do you like best?</a></li>
    <li><a href="/polls/3/">Where do you live?</a></li>
</ul>
```

애플리케이션 개발하기 –View 및 Template

뷰 함수 detail() 및 템플릿 작성

- 3개의 질문 중 하나를 선택했을 때, 질문에 대한 답변 항목을 보여주고 투표하도록 하는 화면
- 템플릿 파일인 detail.html



애플리케이션 개발하기 –View 및 Template

뷰 함수 detail() 및 템플릿 작성

- 템플릿 파일인 detail.html

```
<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>
```

```
<h1>What is your hobby?</h1>


<form action="/polls/1/vote/" method="post">
<input type="hidden" name="csrfmiddlewaretoken" value="NZOayheeZsWueLPZCnyxsZGteBtF6ff7Q5lr1emQAqcWHLsxu2KAq8o4gwBVtvhG">

    <input type="radio" name="choice" id="choice1" value="1" />
    <label for="choice1">Reading</label><br />

    <input type="radio" name="choice" id="choice2" value="2" />
    <label for="choice2">Soccer</label><br />

    <input type="radio" name="choice" id="choice3" value="3" />
    <label for="choice3">Climbing</label><br />

<input type="submit" value="Vote" />
</form>
```

애플리케이션 개발하기 –View 및 Template

뷰 함수 detail() 및 템플릿 작성

- polls₩views.py에 detail() 함수 작성

```
def detail(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'polls/detail.html', {'question': question})
```

애플리케이션 개발하기 –View 및 Template

뷰 함수 vote() 및 리다이렉션 작성

- vote() 뷰 함수의 호출과 연계된 URL은 detail.html 템플릿 파일에서 받음
- vote 버튼을 누르면 vote() 뷰 함수가 호출되는 것

애플리케이션 개발하기 –View 및 Template

뷰 함수 vote() 및 리다이렉션 작성

- views.py파일을 열고 vote() 뷰 함수의 내용 입력

```
def vote(request, 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):
        # Redisplay the question voting form.
        return render(request, 'polls/detail.html', {
            'question': question,
            'error_message': "You didn't select a choice.",
        })
    else:
        selected_choice.votes += 1
        selected_choice.save()
        # Always return an HttpResponseRedirect after successfully dealing
        # with POST data. This prevents data from being posted twice if a
        # user hits the Back button.
        return HttpResponseRedirect(reverse('polls:results', args=(question.id,)))
```

애플리케이션 개발하기 –View 및 Template

뷰 함수 results() 및 리다이렉션 작성

- results()뷰 함수의 호출과 연계된 URL은 votes() 뷰 함수의 리다이렉트 결과
- 폼 데이터를 처리한 후에 그 결과를 보여주는 페이지로 리다이렉트시켜주기 위해 votes() 뷰 함수에 실행

애플리케이션 개발하기 –View 및 Template

뷰 함수 results() 및 리다이렉션 작성

- views.py 파일을 열고 results() 뷰 함수의 내용을 추가

```
def results(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'polls/results.html', {'question': question})
```

애플리케이션 개발하기 –View 및 Template

뷰 함수 results() 및 리다이렉션 작성

- 템플릿은 투표 결과로 각 질문마다 투표 카운트를 보여주는 화면

- 템플릿 파일 **results.html**

```
<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>
```

```
<h1>What is your hobby?</h1>
<ul>
    <li>Reading -- 3 votes</li>
    <li>Soccer -- 1 vote</li>
    <li>Climbing -- 1 vote</li>
</ul>
<a href="/polls/1/">Vote again?</a>
```

애플리케이션 개발하기 –View 및 Template

작업 확인하기

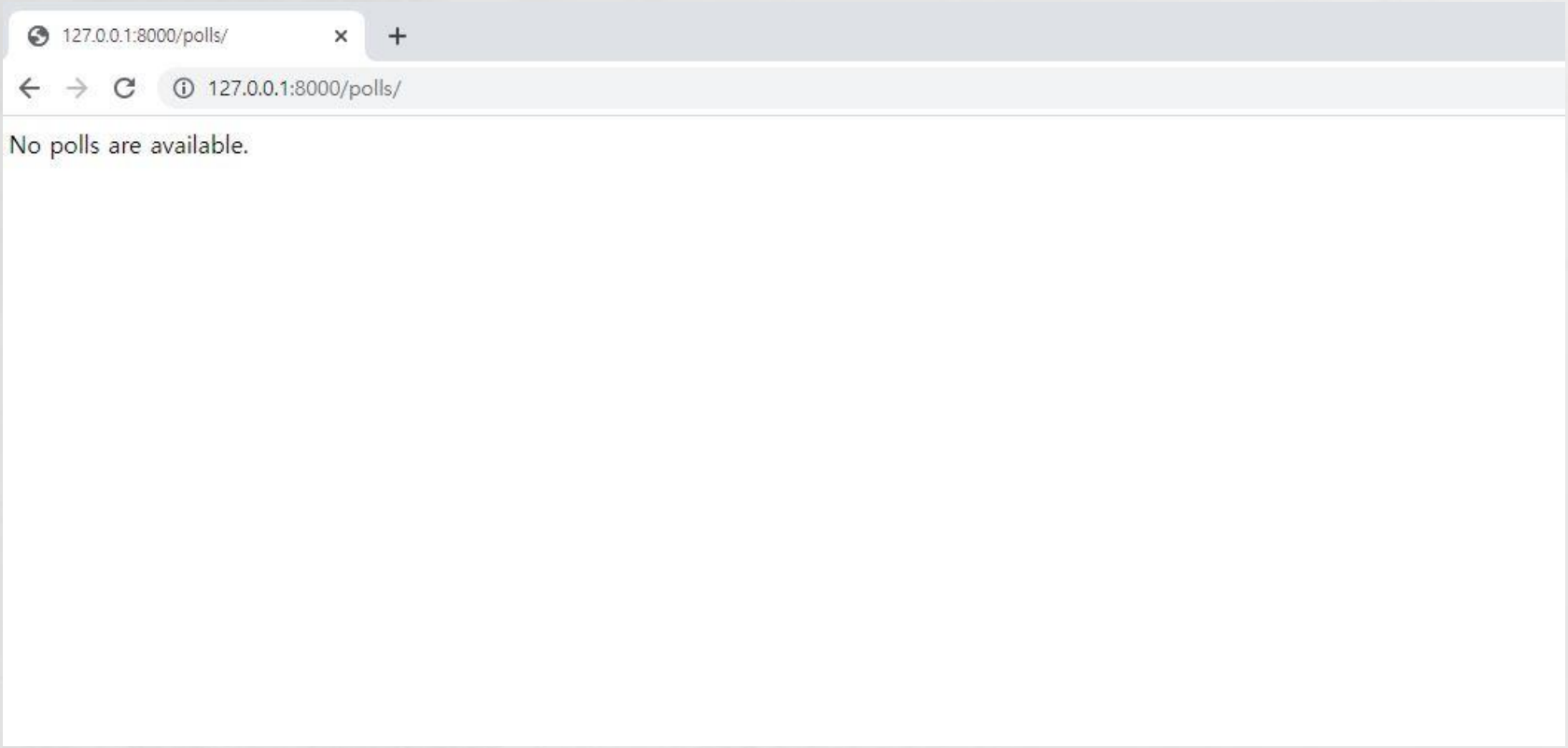
- 웹 서버를 실행하기 위해 하나의 창에서 작업해도 되지만 runserver 용으로 별도의 cmd창을 열어 사용하는 것이 편리
  - (base) C:₩MyTest₩projectsite>python manage.py runserver 127.0.0.1:8000
  - (base) C:₩MyTest₩projectsite>python manage.py runserver
  - (base) C:₩MyTest₩projectsite>python manage.py runserver 0:8000

```
(base) C:₩MyTest>cd projectsite

(base) C:₩MyTest₩projectsite>python manage.py runserver 127.0.0.1:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 23, 2021 - 23:40:17
Django version 3.2.6, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

애플리케이션 개발하기 –View 및 Template

작업 확인하기

- 웹 브라우저에서 http://127.0.0.1:8000/polls

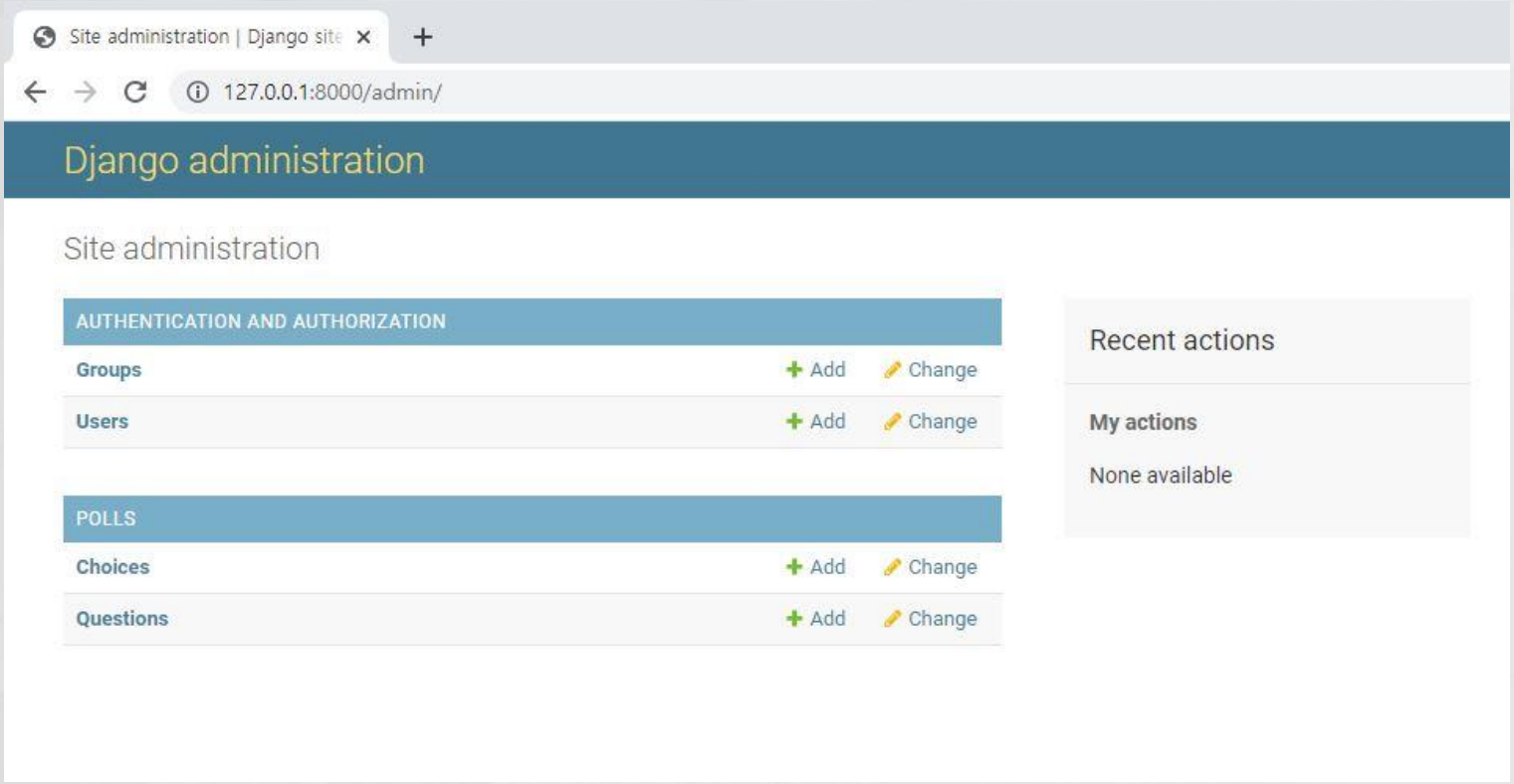


애플리케이션 개발하기 –View 및 Template

작업 확인하기

- 데이터베이스에 데이터가 들어있지 않아서 질문이 없는 빈 페이지

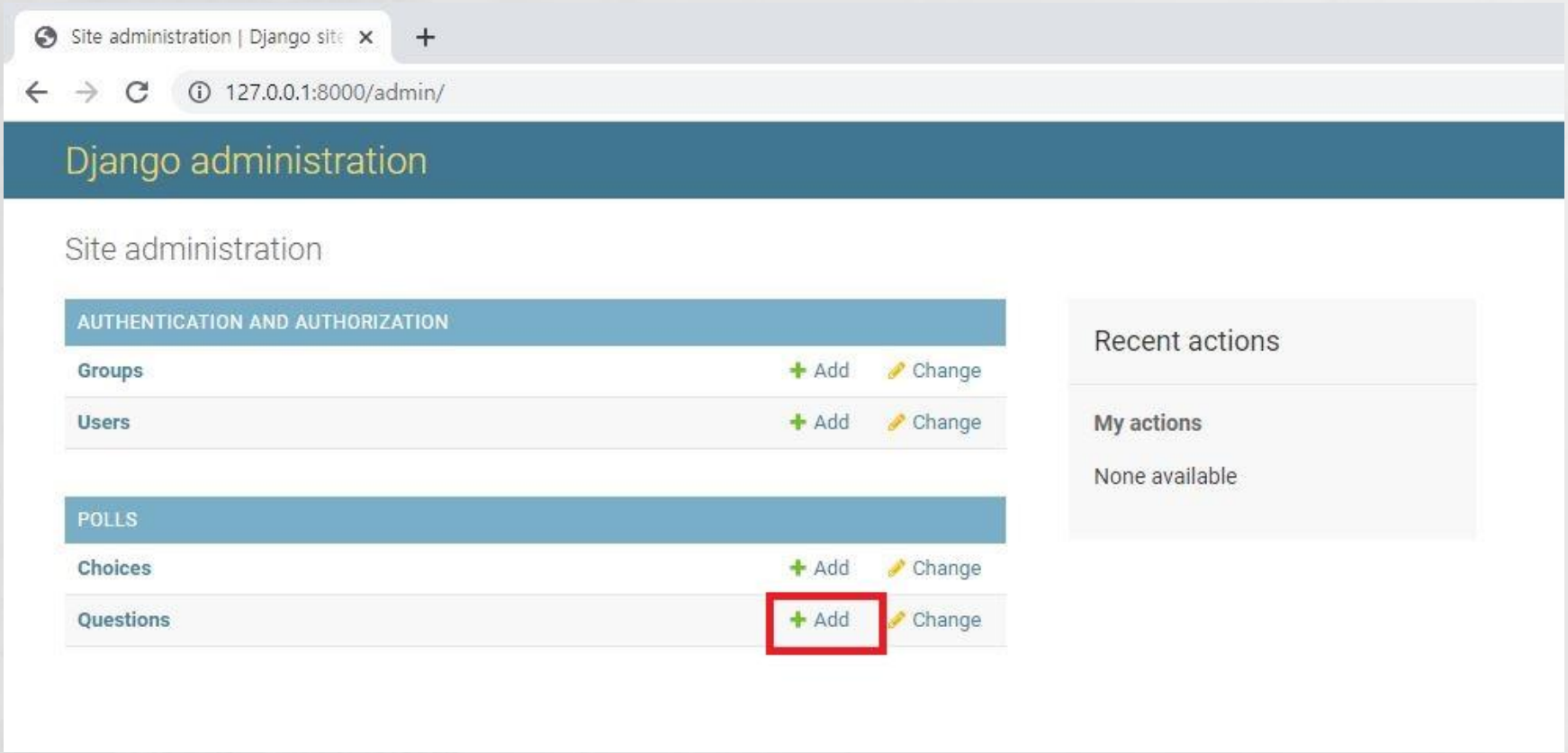
- 웹 브라우저에서 http://127.0.0.1:8000/admin



애플리케이션 개발하기 –View 및 Template

작업 확인하기

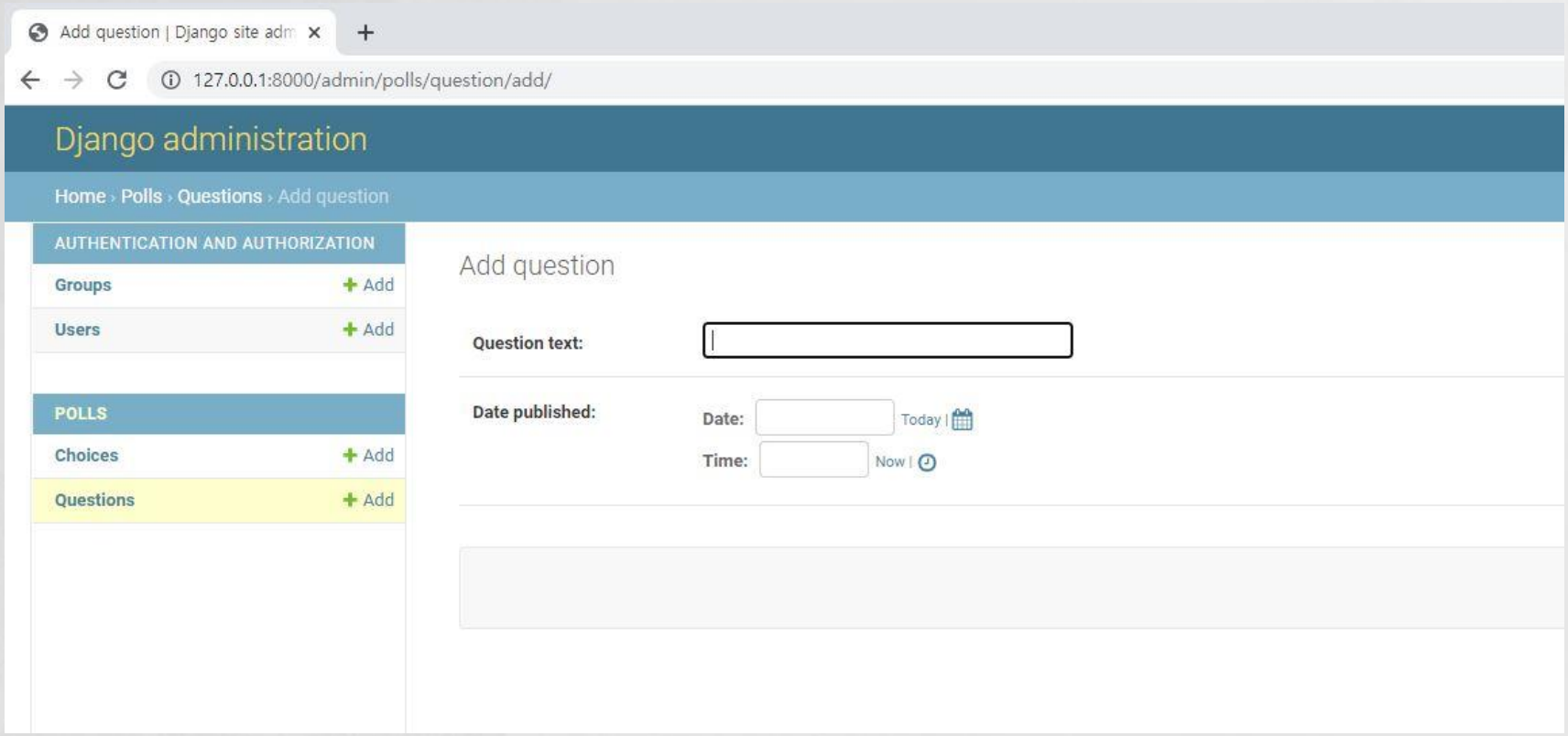
- Questions 데이터 입력



애플리케이션 개발하기 –View 및 Template

작업 확인하기

- Questions 데이터 입력



애플리케이션 개발하기 –View 및 Template

작업 확인하기

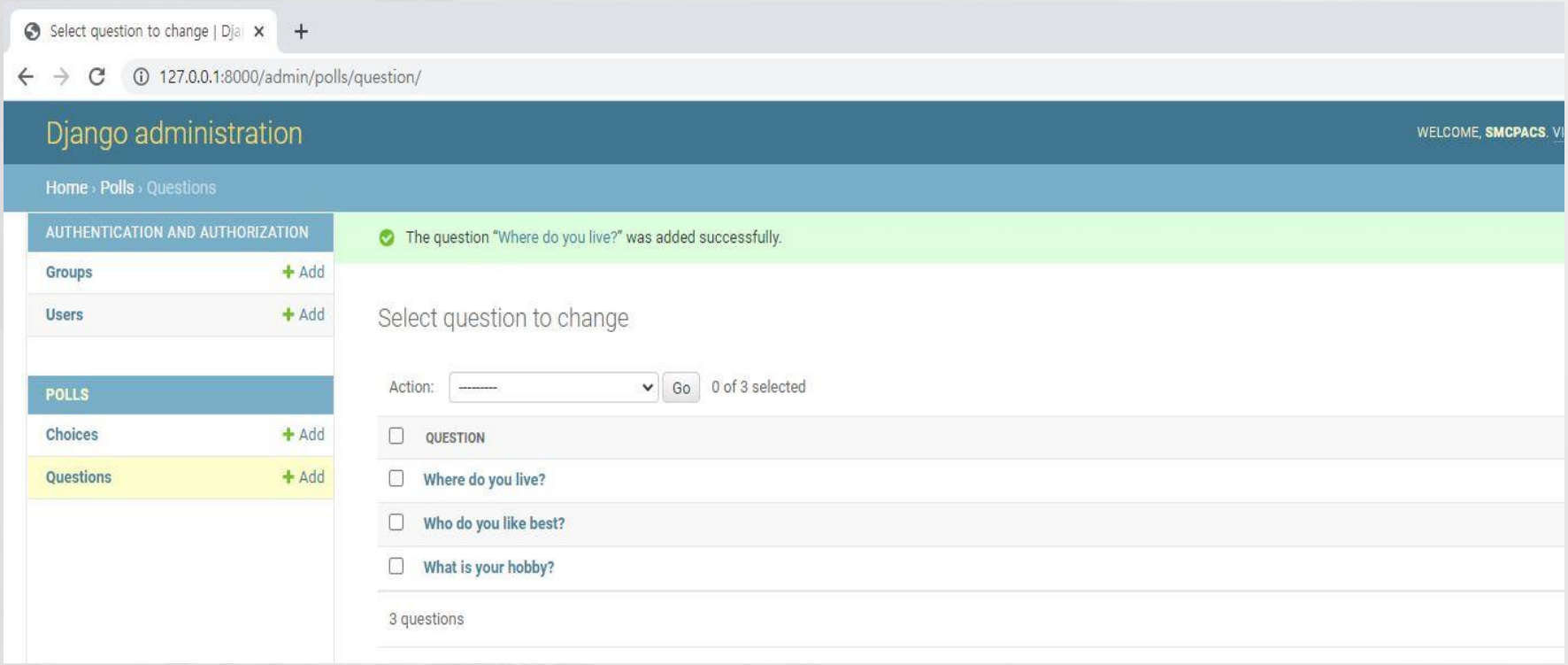
- Questions 데이터 입력

Question Text	Date	Time
What is your hobby?	Today	Now
Who do you like best?	2021-08-25	Midnight
Where do you live?	2021-08-24	Noon

애플리케이션 개발하기 –View 및 Template

작업 확인하기

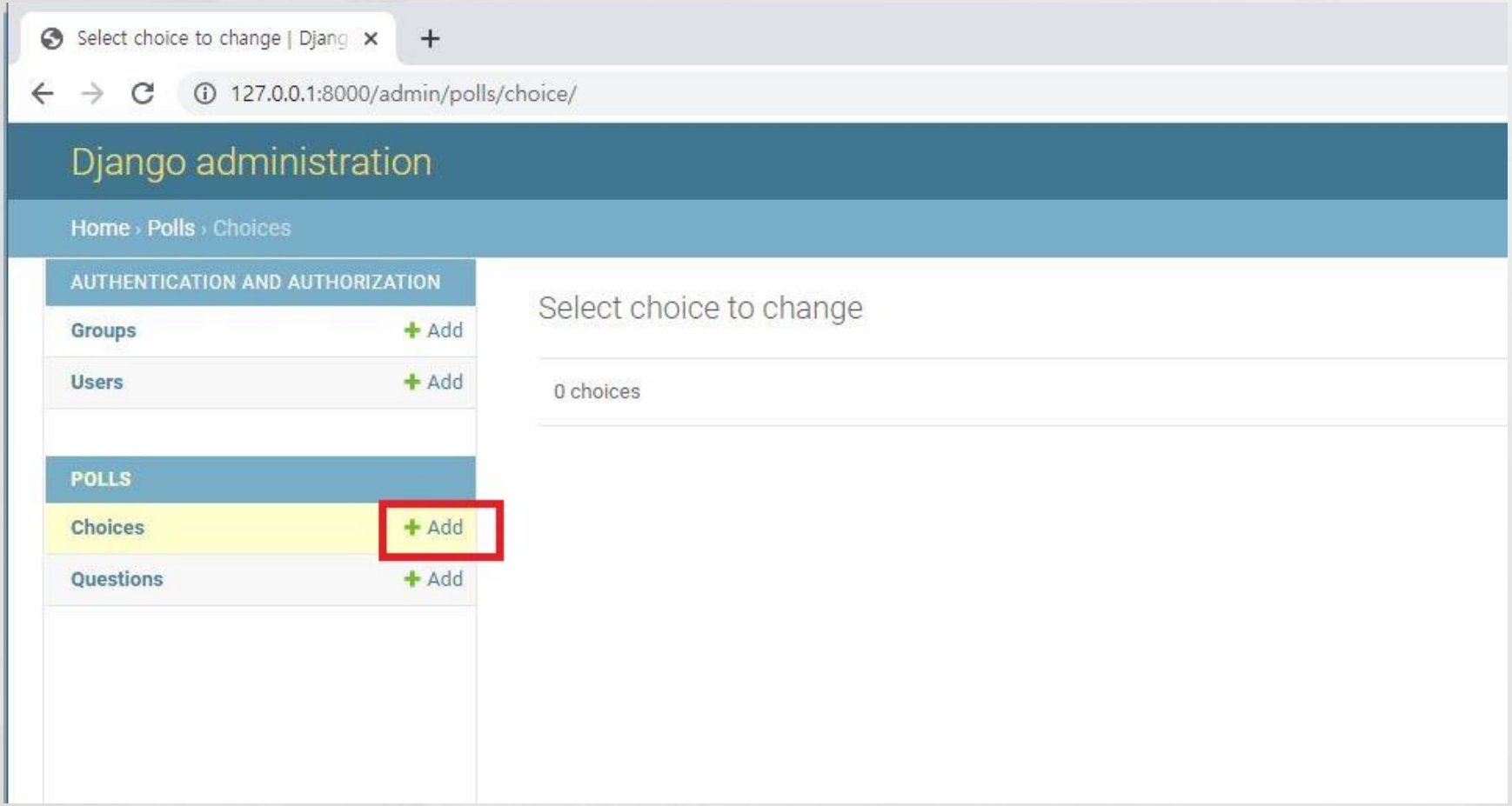
- Questions 데이터 입력



애플리케이션 개발하기 –View 및 Template

작업 확인하기

- Choices 데이터 입력



애플리케이션 개발하기 –View 및 Template

작업 확인하기

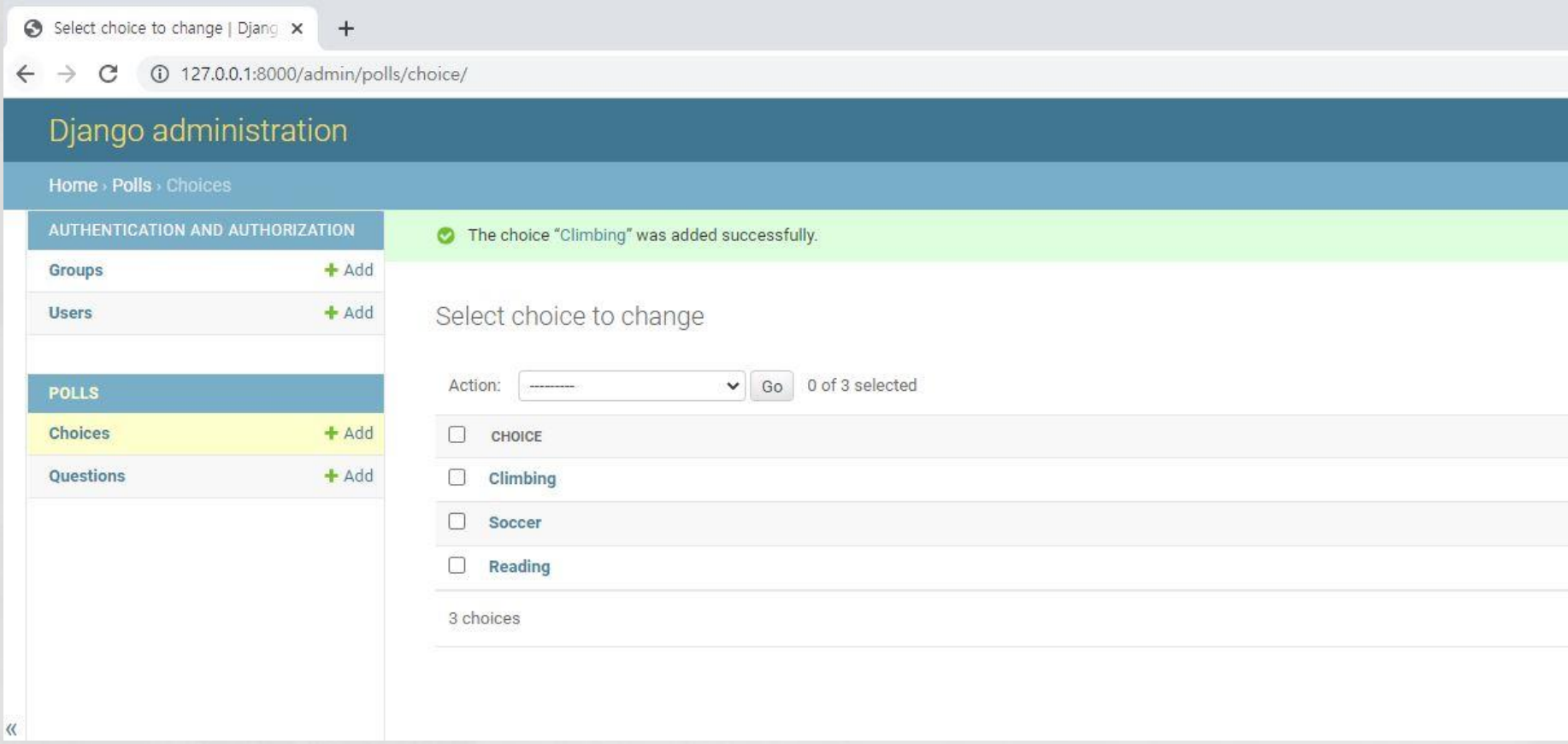
- Choices 데이터 입력

Question	Choice Test	Votes
What is your hobby?	Reading	0
What is your hobby?	Soccer	0
What is your hobby?	Climbing	0

애플리케이션 개발하기 –View 및 Template

작업 확인하기

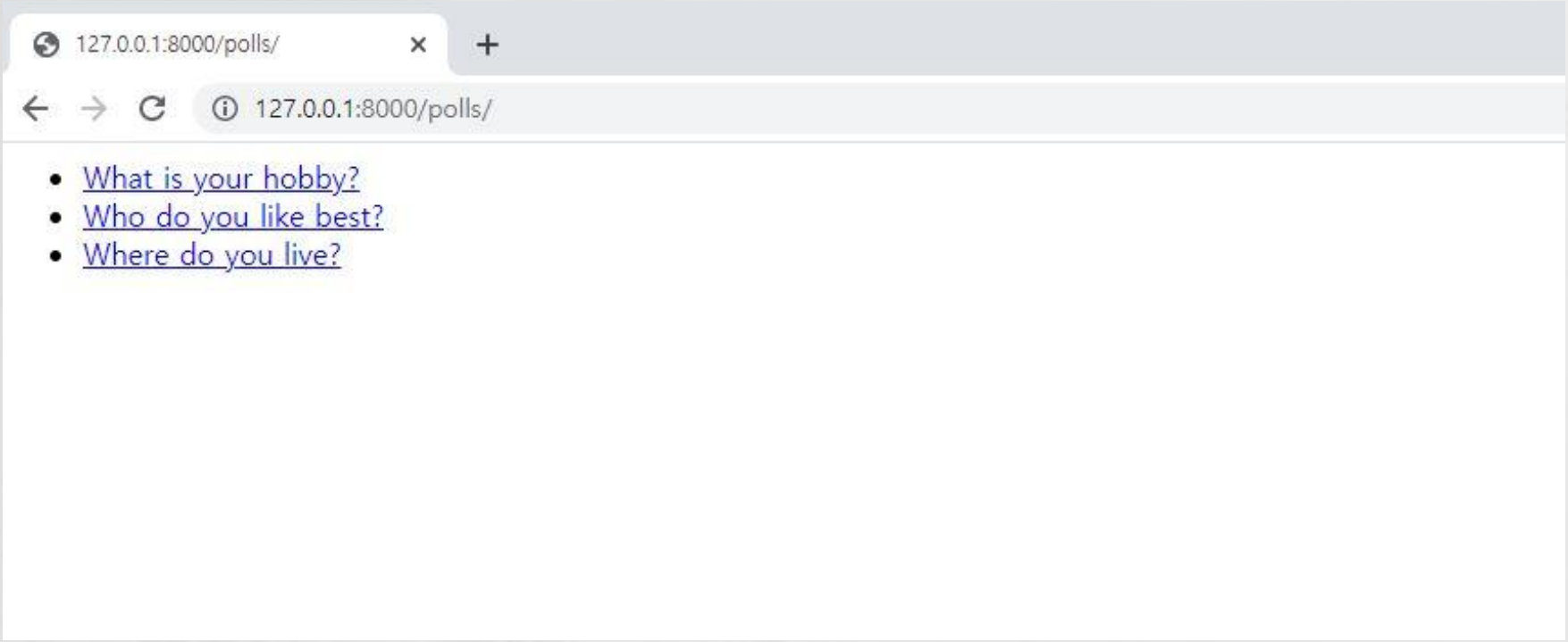
- Choices 데이터 입력



애플리케이션 개발하기 –View 및 Template

작업 확인하기

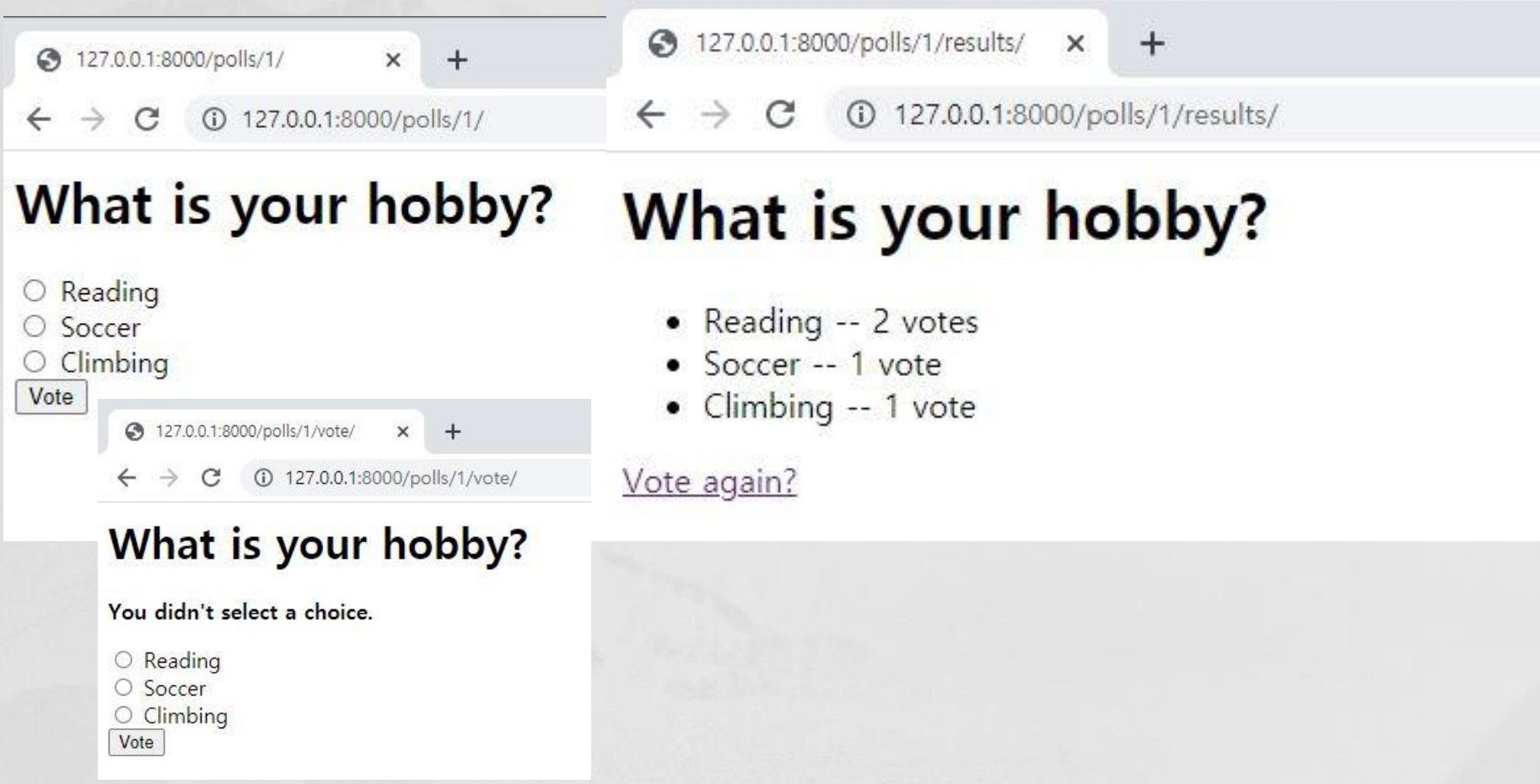
- 웹 브라우저에서 http://127.0.0.1:8000/polls



애플리케이션 개발하기 –View 및 Template

작업 확인하기

- 웹 브라우저에서 http://127.0.0.1:8000/polls



정리

정리

- 애플리케이션 개발하기 - Model 코딩
- 애플리케이션 개발하기 - View 및 Template