

◆ 장고를 이용한(인공지능)웹 앱 구현을 위한 개발 절차

10단계까지는 세팅단계라고 보면 된다

1. 장고 프로젝트 생성
2. settings.py 설정
3. 기본 테이블 생성(migrate)
4. super user 생성 – 반드시 도스프롬프트상에서 가능(해당프로젝트폴더의 manage.py 가 있는 폴더에서 python manage.py createsuperuser
5. app 만들기 createapplication xxxxx
6. settings.py 에 xxxxx.apps.xxxxxConfig 추가
7. xxxxx/models.py 정의
8. xxxxx 폴더에 admin.py 작성
9. 데이터베이스 변경 --- 1. makemigrations xxxxx 2. migrate
10. Admin 으로 접속후 제반 점검 – 필요하면 기초 데이터 등록
11. 프로젝트의 urls.py 작성추가 (2 가지방법)
12. template 단(자바 스프링으로 말하며 view 단) 작성

표 2-3 북마크 앱 – 작업/코딩 순서

작업 순서	관련 명령/파일	필요한 작업 내용
빠대 만들기	startproject	mysite 프로젝트 생성
	settings.py	프로젝트 설정 항목 변경
	migrate	User/Group 테이블 생성
	createsuperuser	프로젝트 관리자인 슈퍼유저를 만들
	startapp	북마크 앱 생성
	settings.py	북마크 앱 등록
모델 코딩하기	models.py	모델(테이블) 정의
	admin.py	Admin 사이트에 모델 등록
	makemigrations	모델의 변경사항 추출
	migrate	변경사항을 데이터베이스에 반영
URLconf 코딩하기	urls.py	URL 정의
뷰 코딩하기	views.py	뷰 로직 작성
템플릿 코딩하기	templates 디렉터리	템플릿 파일 작성
그 외 코딩하기	-	(없음)

[인공지능] 인공지능플랫폼내외부 인터페이스 장고개발의 일반적 절차 <주원쌤 다음카페 참조>

1. 장고 프로젝트 생성

프로젝트 만들고 나서 Migrate, Custom command 해주기!!!!!!!!!!!!

2. settings.py 설정 부분

```
1 from pathlib import Path
2 import os
3
4 BASE_DIR = Path(__file__).resolve().parent.parent
5
6 # SECURITY WARNING: keep the secret key used in production secret!
7 # SECRET_KEY는 만들어진 그대로 두고 사용할 것
8 SECRET_KEY = 'ds@a8eqqzcjwzg=%7mt-)%royhpj)i4mk76w6cis1_2hph&q5'
9 # SECURITY WARNING: don't run with debug turned on in production!
10
11 # True를 세팅하면 개발 모드, False를 세팅하면 운영 모드
12 # False인 운영모드인 경우 아래 ALLOWED_HOSTS 리스트에 반드시 서버의 IP 혹은 도메인을 지정해야 함
13 # True인 개발 모드인 경우는 'localhost', '127.0.0.1'로 맞추함
14 DEBUG = True
15 ALLOWED_HOSTS = []
16
17
18 # Application definition
19 # 새로운 앱이 만들어지면 이 자리에 등록해야 함
20 INSTALLED_APPS = [
21     'django.contrib.admin',
22     'django.contrib.auth',
23     'django.contrib.contenttypes',
24     'django.contrib.sessions',
25     'django.contrib.messages',
26     'django.contrib.staticfiles',
27 ]
28
29 MIDDLEWARE = [
30     'django.middleware.security.SecurityMiddleware',
31     'django.contrib.sessions.middleware.SessionMiddleware',
32     'django.middleware.common.CommonMiddleware',
33     'django.middleware.csrf.CsrfViewMiddleware',
34     'django.contrib.auth.middleware.AuthenticationMiddleware',
35     'django.contrib.messages.middleware.MessageMiddleware',
36     'django.middleware.clickjacking.XFrameOptionsMiddleware',
37 ]
38
39 ROOT_URLCONF = 'mysite.urls'
40
```

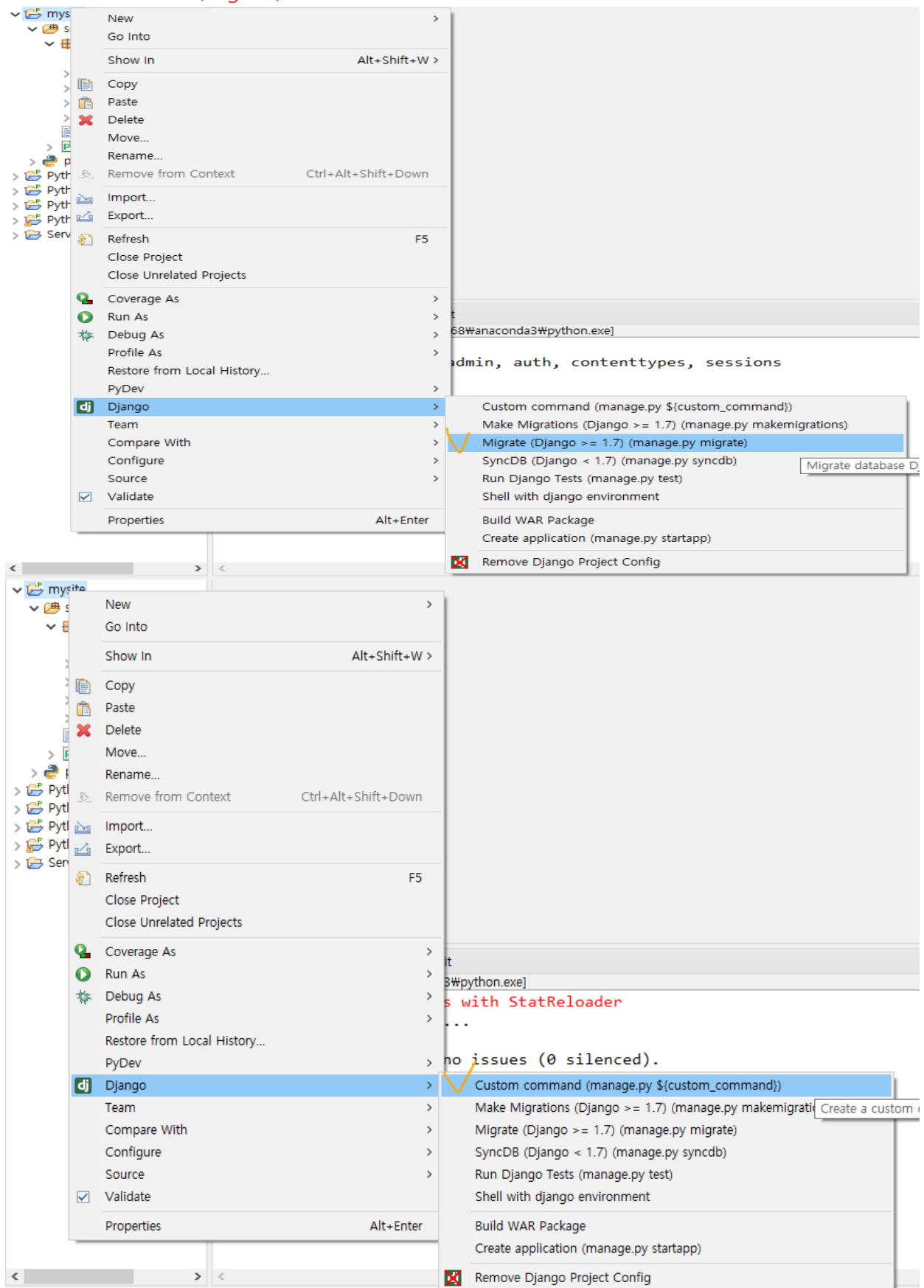
```
41 # 프로젝트의 템플릿 파일이 위치할 디렉토리를 지정 (MVT == 스프링의 MVC 모델)
42 TEMPLATES = [
43     {
44         'BACKEND': 'django.template.backends.django.DjangoTemplates',
45         'DIRS': [os.path.join(BASE_DIR, 'templates')], #os 패키지를 임포트해야함
46         'APP_DIRS': True,
47         'OPTIONS': {
48             'context_processors': [
49                 'django.template.context_processors.debug',
50                 'django.template.context_processors.request',
51                 'django.contrib.auth.context_processors.auth',
52                 'django.contrib.messages.context_processors.messages',
53             ],
54         },
55     ],
56 ]
57
58 WSGI_APPLICATION = 'mysite.wsgi.application'
59
60
61 # Database
62 # https://docs.djangoproject.com/en/3.1/ref/settings/#databases
63
64 DATABASES = {
65     'default': {
66         'ENGINE': 'django.db.backends.sqlite3',
67         'NAME': BASE_DIR / 'db.sqlite3', #src폴더안에 생긴 db.sqlite3
68     }
69 }
70
71
```

db.sqlite3는 프로젝트 생성 후 migrate를 할 때 생성되는 것임!!!!!!!!!!

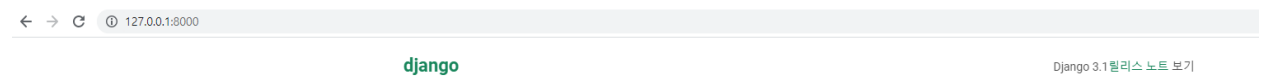
```
72 # Password validation
73 # https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators
74
75 AUTH_PASSWORD_VALIDATORS = [
76     {
77         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
78     },
79     {
80         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
81     },
82     {
83         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
84     },
85     {
86         'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
87     },
88 ]
89
90
91 # Internationalization
92 # https://docs.djangoproject.com/en/3.1/topics/i18n/
93
94 LANGUAGE_CODE = 'ko-kr' #'en-us'
95
96 TIME_ZONE = 'Asia/Seoul' #'UTC'
97
98 USE_I18N = True
99
100 USE_L10N = True
101
102 USE_TZ = True #타임존을 사용하지 않으면 에러를 일으킴
103
```

```
104
105 # Static files (CSS, JavaScript, Images)
106 # https://docs.djangoproject.com/en/3.1/howto/static-files/
107
108 # 참고가 자동으로 만든 static들의 경로
109 STATIC_URL = '/static/'
110 # 우선적으로 써야할 것이 아래와 같이 statics 임의의 경로를 추가해서 사용할 수 있다
111 # STATICFILES_DIRS = [os.path.join(BASE_DIR, 'static0')]
112
113 # 미디어 부분은 따로 추가 할 수 있다 (추가한 부분임)
114 MEDIA_URL = '/media/'
115 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

3. 기본 테이블 생성(migrate) 단계



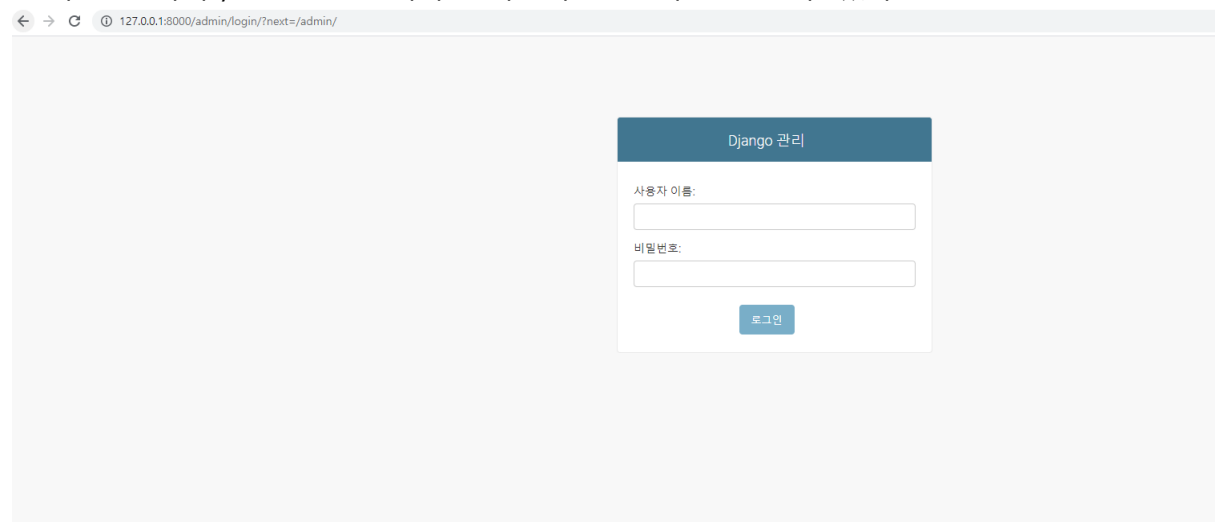
🔗 브라우저에서 실행하면 한글로 바뀐 것 확인가능



성공적으로 설치되었습니다! 축하합니다!

이 페이지는 어떤 URL도 지정되지 않았고, settings 파일에 `DEBUG=True`가 설정되어 있을 때 표시됩니다.

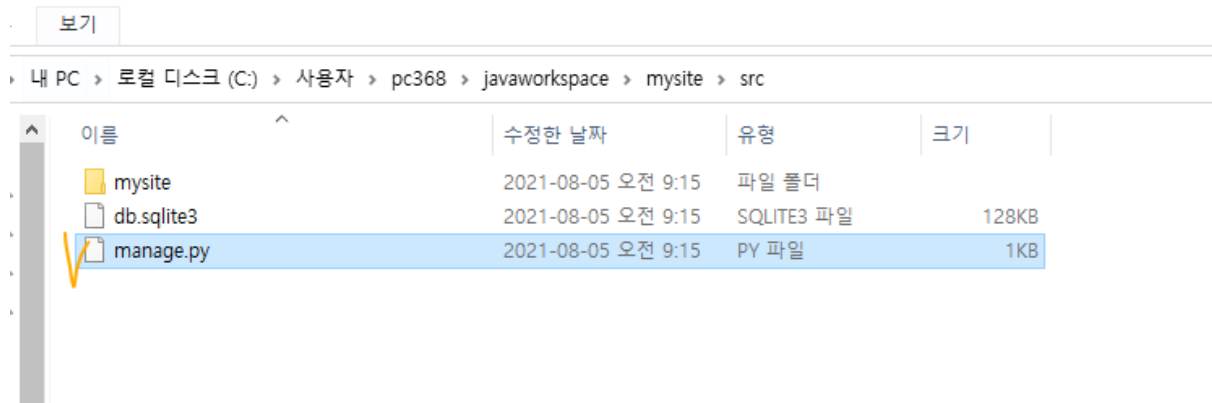
🔗 주소창 뒤에 /admin 을 입력하면 다음과 같은 화면을 볼 수 있다!!



4. super user 생성 – 반드시 도스프롬프트상에서 가능(해당프로젝트폴더의 manage.py 가 있는 폴더에서 python manage.py createsuperuser

지금 작업하고 있는 프로젝트 소스에 들어가서 manage.py가 있는 경로를 복사하기

(현재 작업하고 있는 컴퓨터에서의 경로 C:\Users\pc368\javaworkspace\mysite\src)



드라이브위치를 바꿀 땐 D에 있으면 d: F에 있으면 f: 이런 식으로 옮겨주고

cd 명령어 다음에 복사한 경로를 붙여 넣고

```
관리자: Anaconda Powershell Prompt (anaconda3)
(base) PS C:\WINDOWS\system32> cd C:\Users\pc368\javaworkspace\mysite\src
(base) PS C:\Users\pc368\javaworkspace\mysite\src> dir

디렉터리: C:\Users\pc368\javaworkspace\mysite\src

Mode                LastWriteTime         Length Name
----                -
d-----          2021-08-05 오전 9:15             mysite
-a-----          2021-08-05 오전 9:15        131072 db.sqlite3
-a-----          2021-08-05 오전 9:15         684 manage.py

(base) PS C:\Users\pc368\javaworkspace\mysite\src> python manage.py createsuperuser
사용자 이름 (leave blank to use 'pc368'): mysite
이메일 주소: mysite@vision.co.kr
Password:
Password (again):
비밀번호가 사용자 이름과 너무 유사합니다.
비밀번호가 너무 짧습니다. 최소 8 문자를 포함해야 합니다.
Bypass password validation and create user anyway? [y/N] y
Superuser created successfully.
(base) PS C:\Users\pc368\javaworkspace\mysite\src>
```

python manage.py createsuperuser 입력해주기

패스워드도 사용자 이름과 동일하게 mysite 를 넣어줬음(일단 통일)

위와 같은 설정이 끝났다면 여기서 입력하고 로그인해본다



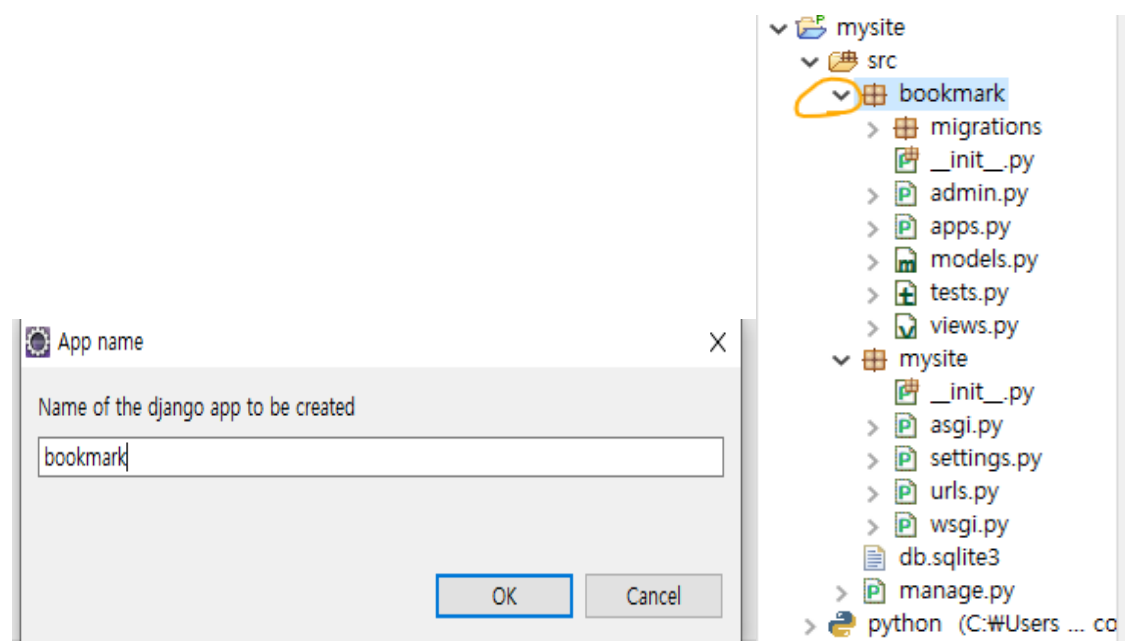
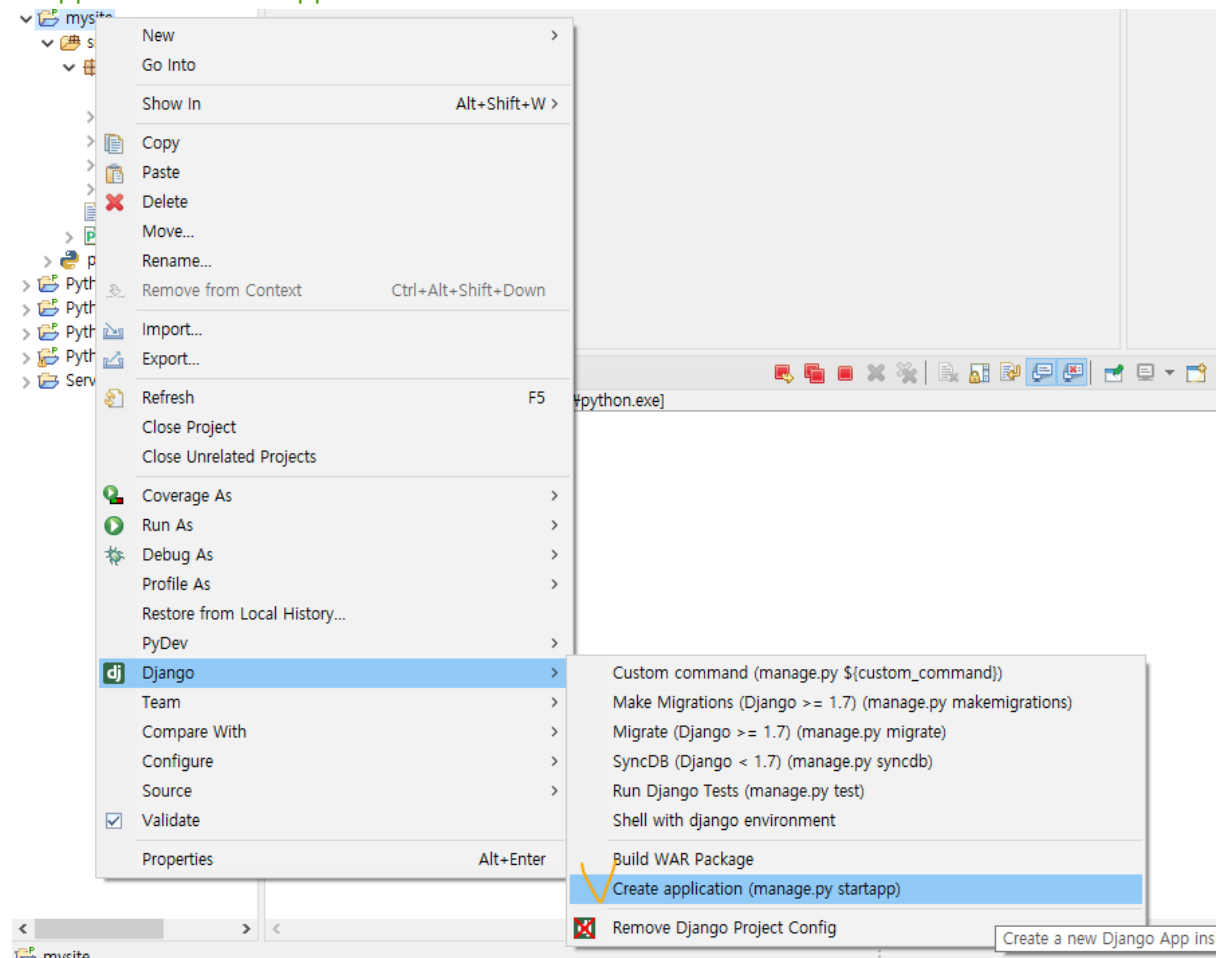
The image shows a Django login form titled "Django 관리". It has two input fields: "사용자 이름:" (Username) with the value "mysite" and "비밀번호:" (Password) with masked characters ".....". Both fields have a yellow checkmark next to them. Below the fields is a blue button labeled "로그인" (Login).

그러면 이러한 화면이 표시된다!



The image shows the Django admin interface. The browser address bar shows "127.0.0.1:8000/admin/". The page title is "Django 관리". The main content area is divided into two columns. The left column is titled "사이트 관리" (Site Management) and contains a table with two rows: "그룹" (Groups) and "사용자(들)" (Users). Each row has a green plus icon for "추가" (Add) and a yellow pencil icon for "변경" (Change). The right column is titled "최근 활동" (Recent Activity) and contains a section "나의 활동" (My Activity) with the text "이용할 수 없습니다." (You cannot use it).

5. app 만들기 createapplication xxxxx



App name 을 지정해주고 새로고침 후 소스를 보면 bookmark 가 생긴 것 확인 가능

6. settings.py 에 xxxxx.apps.xxxConfig 추가(xxxxx 는 만든 App name 을 넣는다)

```
settings.py admin models
11 # True로 세팅하면 개발모드, False로 세팅하면 운영모드
12 # False인 운영모드인 경우 아래 ALLOWED_HOSTS 리스트에 반드시 서버의 ip 혹은 도메인을 지정해야 함
13 # True인 개발 모드인 경우는 'localhost', '127.0.0.1'로 간주함
14 DEBUG = True
15 ALLOWED_HOSTS = []
16
17
18 # Application definition
19 # 새로운 앱이 만들어지면 이 자리에 등록해야 함
20 INSTALLED_APPS = [
21     'django.contrib.admin',
22     'django.contrib.auth',
23     'django.contrib.contenttypes',
24     'django.contrib.sessions',
25     'django.contrib.messages',
26     'django.contrib.staticfiles',
27     'bookmark.apps.BookmarkConfig',
28 ]
29
```

☞ 뒤에 심표(,) 주의!!

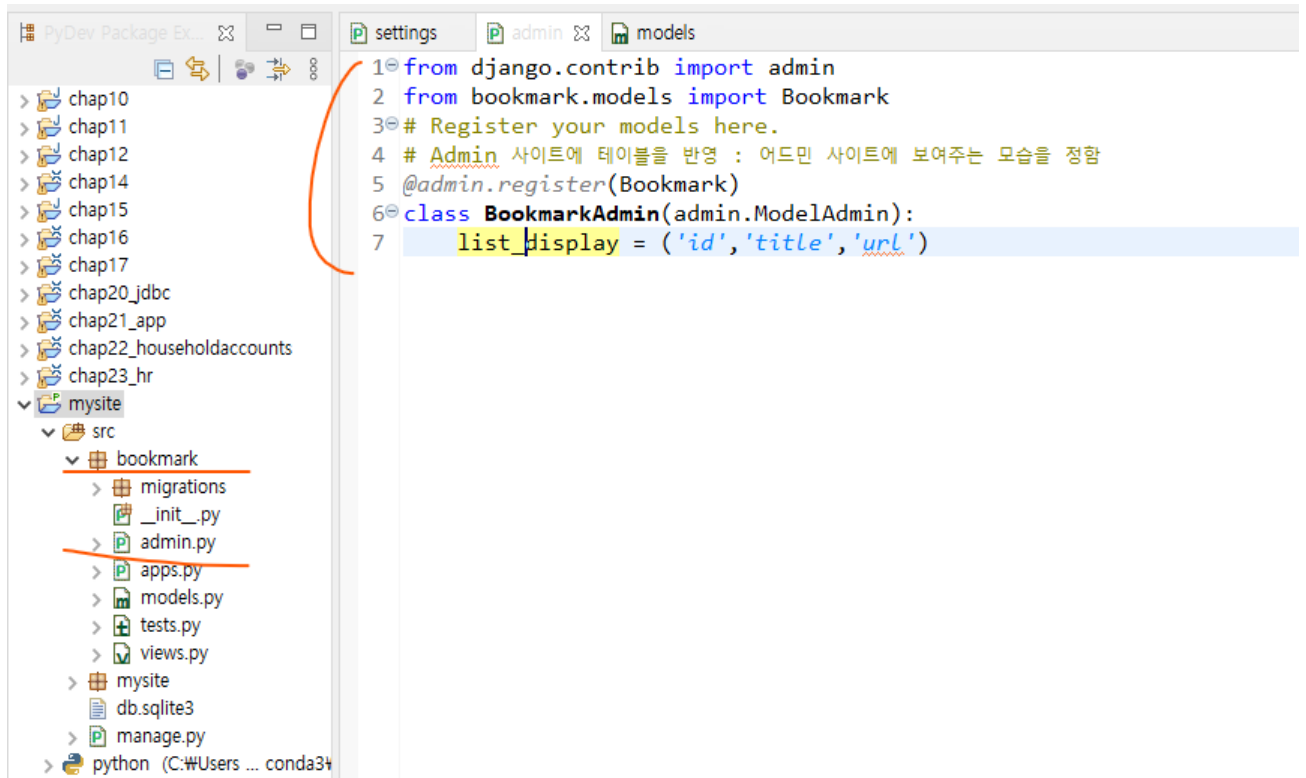
7. xxxxx/models.py 정의(bookmark/models.py)

만들어진 bookmark 밑에서 정의

```
PyDev Package Explorer settings models
> chap10
> chap11
> chap12
> chap14
> chap15
> chap16
> chap17
> chap20_jdbc
> chap21_app
> chap22_householdaccounts
> chap23_hr
▼ mysite
  ▼ src
    ▼ bookmark
      > migrations
      > _init_.py
      > admin.py
      > apps.py
      > models.py
      > tests.py
      > views.py
    > mysite
    > db.sqlite3
    > manage.py
  > python (C:\Users\...conda34)

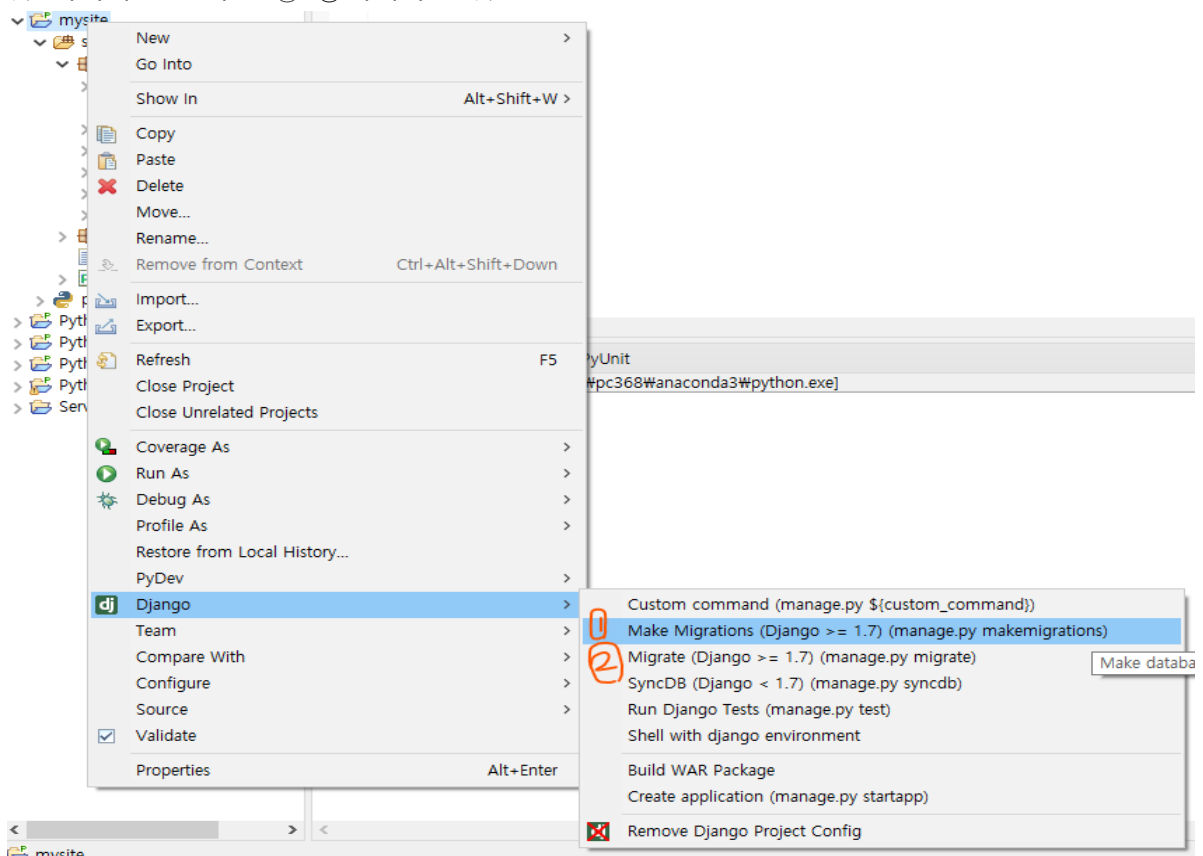
1 from django.db import models
2
3 # Create your models here.
4 # <class> 클래스
5 class Bookmark(models.Model):
6     title = models.CharField('TITLE', max_length=100, blank=True) # models.CharField = 오라클의 VARCHAR2 #verbose_name
7     url = models.URLField('URL', unique=True)
8
9     def __str__(self): #객체표현 양식
10         return self.title
11
```

8. xxxxx 폴더에 admin.py 작성(bookmark 폴더)

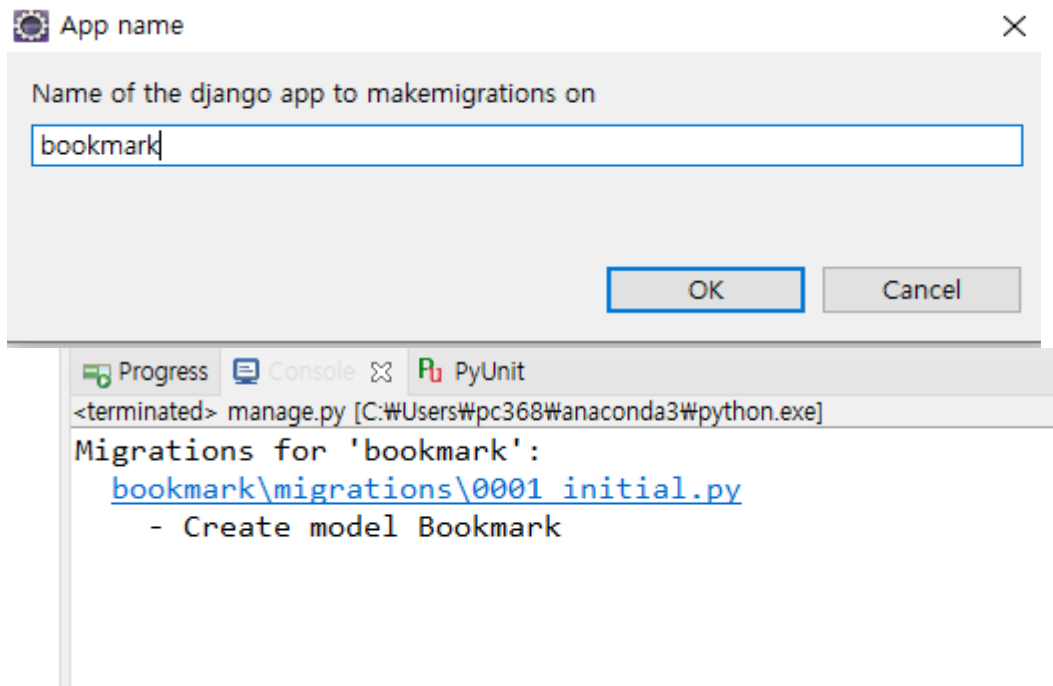


9. 데이터베이스 변경 --- 1. makemigrations xxxxx 2. migrate

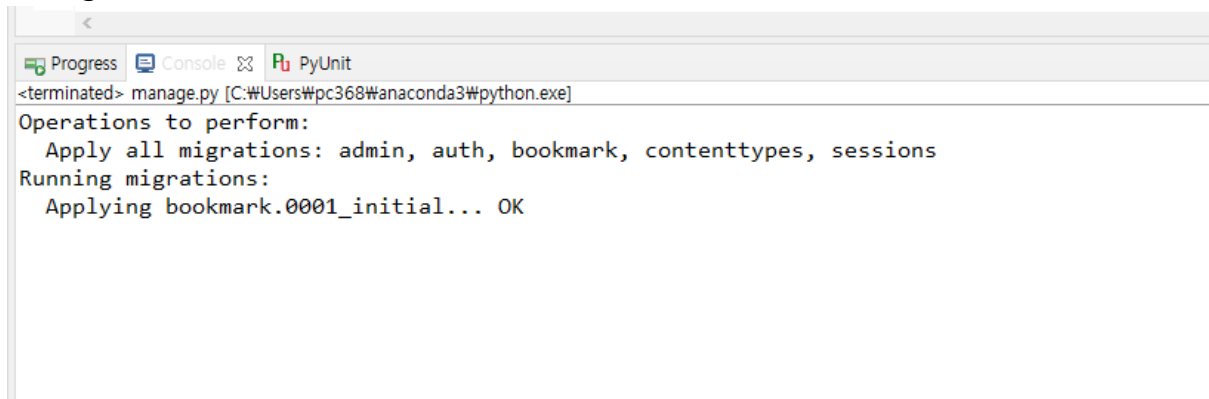
★ 디비작업을 하면 ① ②해줘야함 ★



① Make Migrations 를 클릭하면 다음과 같이 App name 지정이 나오는데 bookmark 넣기



② Migrate 해주기



10. Admin 으로 접속 후 제반 점검 - 필요하면 기초 데이터 등록

← → ↺ 127.0.0.1:8000/admin/

Django 관리

사이트 관리

BOOKMARK		
Bookmarks	+ 추가	변경
인증 및 권한		
그룹	+ 추가	변경
사용자(들)	+ 추가	변경

최근 활동

나의 활동

이용할 수 없습니다.

Django 관리

환영합니다. MY SITE 사이트 보기 / 비밀번호로 변경 / 로그인

홈 > Bookmark > Bookmarks > bookmark 추가

BOOKMARK

Bookmarks + 추가

인증 및 권한

그룹 + 추가

사용자(들) + 추가

bookmark 추가

TITLE: 우리들의 카페

URL: cafe.daum.net/ncsit

저장 및 다른 이름으로 추가 저장 및 편집 계속 저장

Django 관리

홈 > Bookmark > Bookmarks

BOOKMARK

Bookmarks + 추가

인증 및 권한

그룹 + 추가

사용자(들) + 추가

Bookmark "우리들의 카페"가 성공적으로 추가되었습니다.

변경할 bookmark 선택

액션: 1 중 아무것도 선택되지 않았습니다.

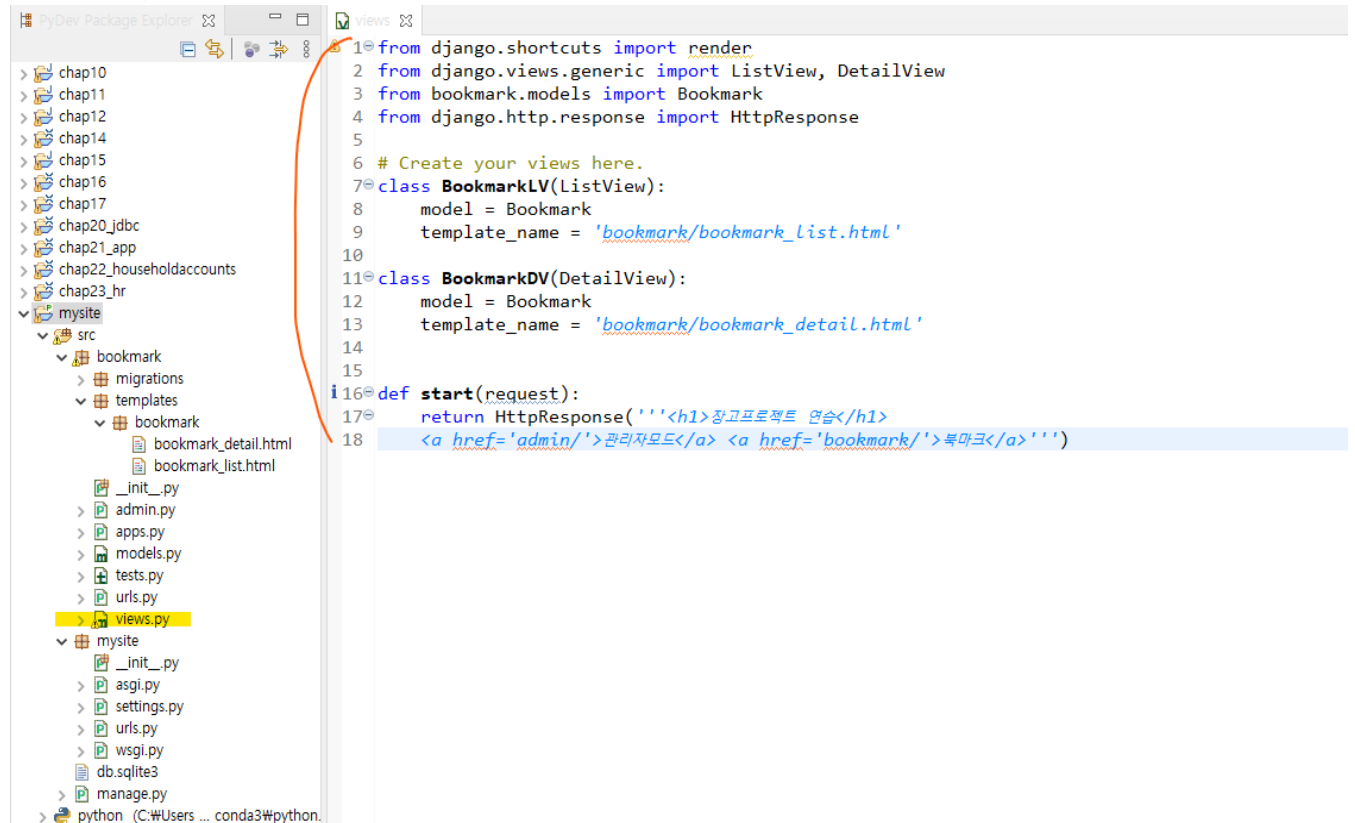
<input type="checkbox"/>	ID	TITLE	URL
<input type="checkbox"/>	1	우리들의 카페	http://cafe.daum.net/ncsit

1 bookmark

11. 프로젝트의 urls.py 작성 추가 (2 가지방법)

방법 ①

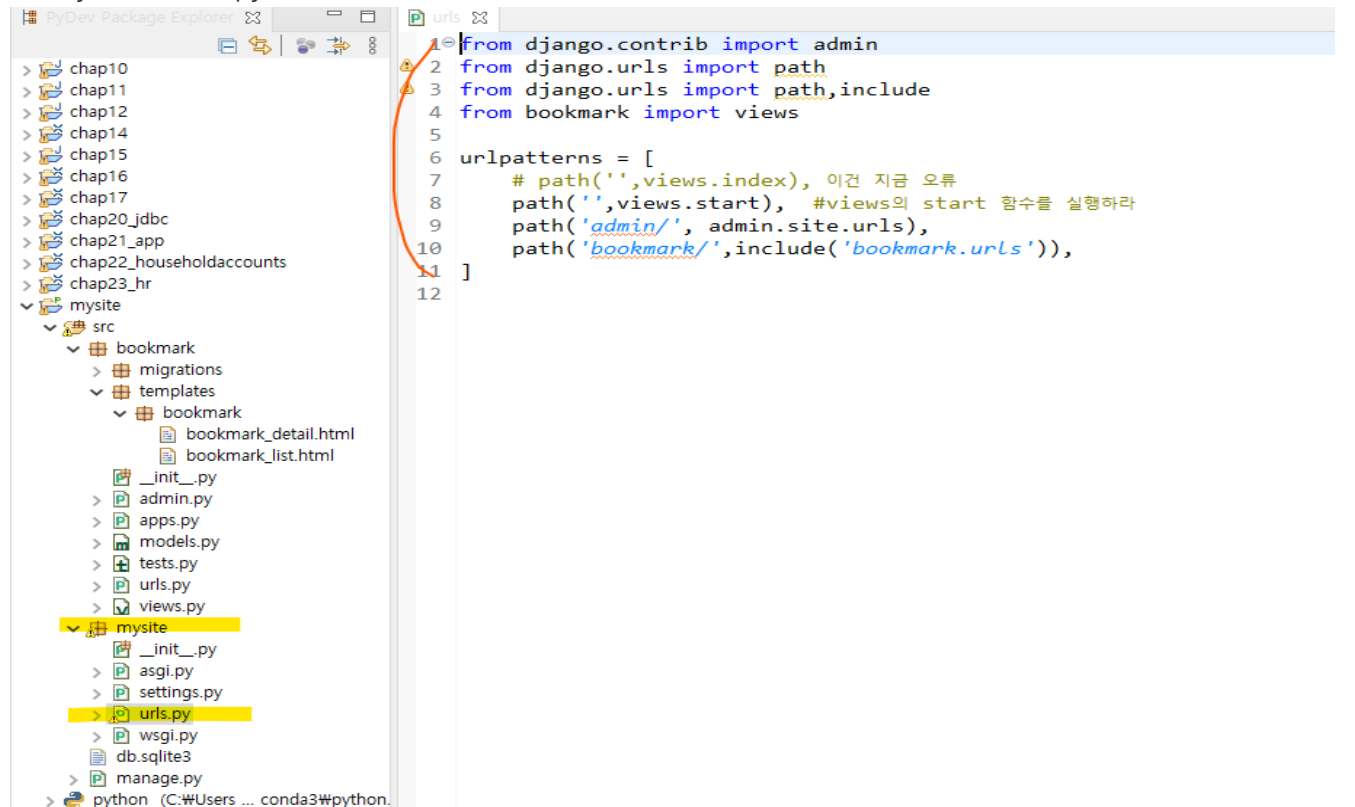
views.py 에서 위와 같이 타이핑



The screenshot shows the PyDev Package Explorer on the left and the views.py file in the editor on the right. The Package Explorer shows a project structure with a 'mysite' folder containing 'src' and 'mysite' subfolders. The 'src' folder contains 'bookmark' and 'mysite' subfolders. The 'bookmark' folder contains 'migrations', 'templates', and 'bookmark' subfolders. The 'bookmark' folder contains 'bookmark_detail.html' and 'bookmark_list.html' files. The 'mysite' folder contains 'admin.py', 'apps.py', 'models.py', 'tests.py', 'urls.py', and 'views.py' files. The 'views.py' file is highlighted. The editor shows the following code:

```
1 from django.shortcuts import render
2 from django.views.generic import ListView, DetailView
3 from bookmark.models import Bookmark
4 from django.http.response import HttpResponse
5
6 # Create your views here.
7 class BookmarkLV(ListView):
8     model = Bookmark
9     template_name = 'bookmark/bookmark_list.html'
10
11 class BookmarkDV(DetailView):
12     model = Bookmark
13     template_name = 'bookmark/bookmark_detail.html'
14
15
16 def start(request):
17     return HttpResponse(''<h1>장고프로젝트 연습</h1>
18     <a href='admin/'>관리자모드</a> <a href='bookmark/'>북마크</a>'')
```

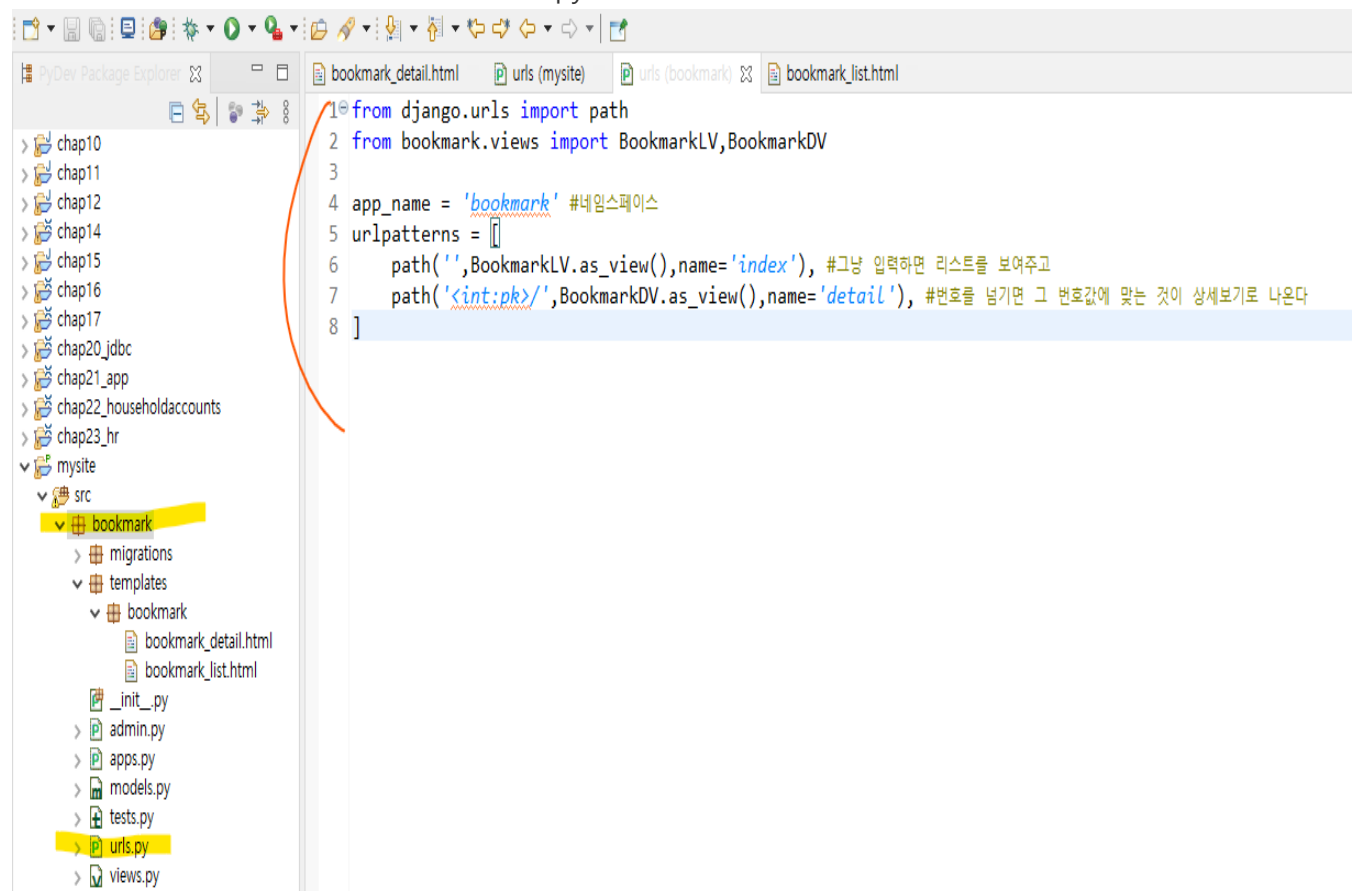
mysite 의 urls.py 에서 타이핑을 추가해준다



The screenshot shows the PyDev Package Explorer on the left and the urls.py file in the editor on the right. The Package Explorer shows the same project structure as the previous screenshot, but with 'urls.py' highlighted in the 'mysite' folder. The editor shows the following code:

```
1 from django.contrib import admin
2 from django.urls import path
3 from django.urls import path,include
4 from bookmark import views
5
6 urlpatterns = [
7     # path('',views.index), 이걸 지금 오류
8     path('',views.start), #views의 start 함수를 실행하라
9     path('admin/', admin.site.urls),
10    path('bookmark/',include('bookmark.urls')),
11 ]
12
```

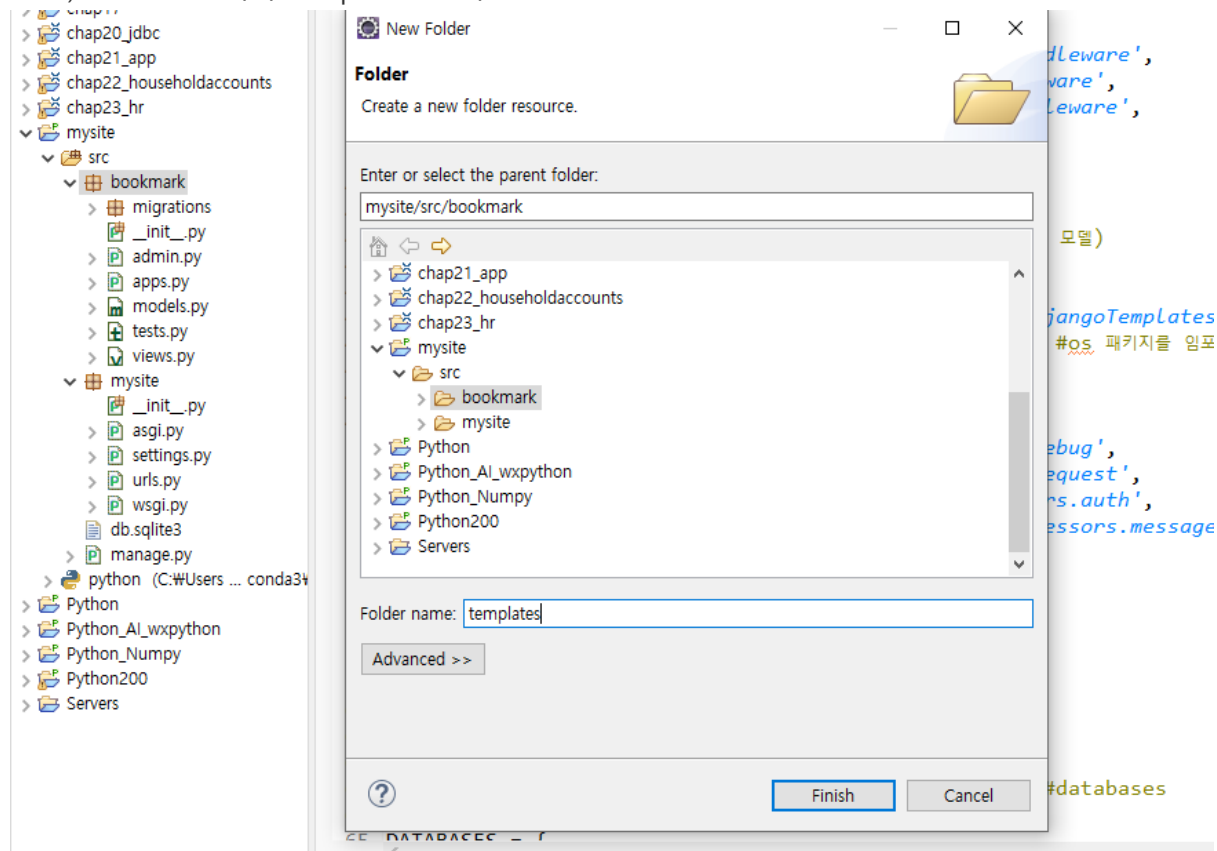
bookmark 에 urls 모듈을 만들고 urls.py 에서 타이핑을 추가해준다



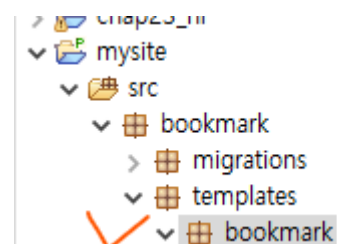
```
1 from django.urls import path
2 from bookmark.views import BookmarkLV, BookmarkDV
3
4 app_name = 'bookmark' #네임스페이스
5 urlpatterns = []
6 path('', BookmarkLV.as_view(), name='index'), #그냥 입력하면 리스트를 보여주고
7 path('<int:pk>', BookmarkDV.as_view(), name='detail'), #번호를 넘기면 그 번호값에 맞는 것이 상세보기로 나온다
8 ]
```

12. template 단(자바 스프링으로 말하며 view 단) 작성

☞ 1) bookmark 에서 templates 폴더 생성



☞ 2) templates에서 bookmark 폴더 생성



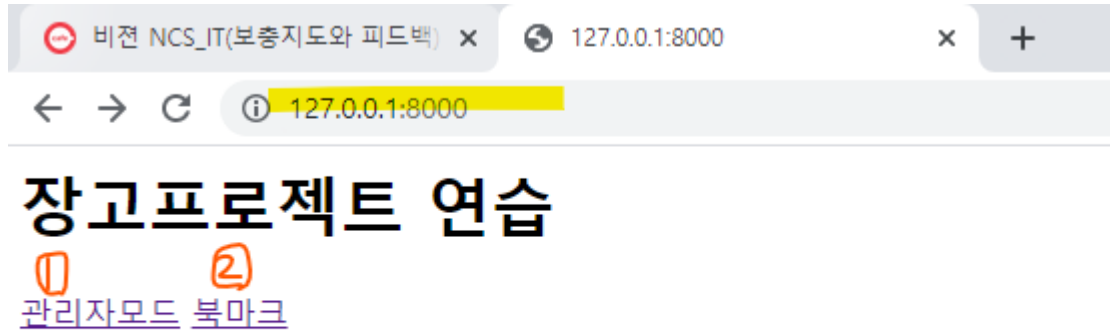
☞ 3) 2)에서 생성한 bookmark에 HTML File 생성(bookmark_list.html, bookmark_detail.html)


```
settings admin models urls bookmark_list.html views
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Django Bookmark List</title>
6 </head>
7 <body>
8 <div id="content">
9     <h1>Bookmark List</h1>
10    <ul>
11        {% for bookmark in object_list %} <!-- 데이터베이스에서 읽어서 object_list로 넘기기 -->
12        <li><a href="{% url 'bookmark:detail' bookmark.id %}">{{bookmark}}</a></li>
13        {% endfor %}
14    </ul>
15
16 </div>
17 </body>
18 </html>
```

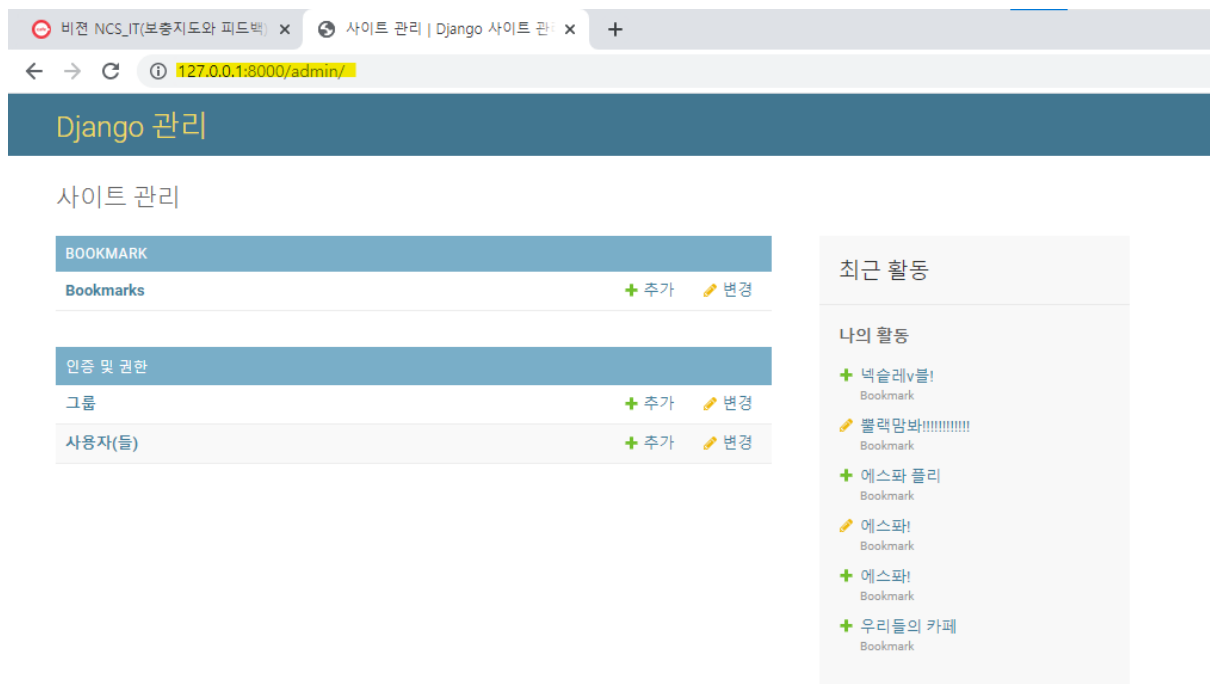
```
bookmark_detail.html urls (mysite) urls (bookmark) bookmark_list.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Django Bookmark Detail</title>
6 </head>
7 <body>
8 <div id="content">
9     <h1>{{object.title}}</h1>
10    <ul>
11        <li><a href="{{object.url}}">{{object.url}}</a></li>
12    </ul>
13 </div>
14 </body>
15 </html>
```

☞ 모든 단계(1-12)를 다하고 서버를 재실행하면 다음과 같은 화면이 나온다.

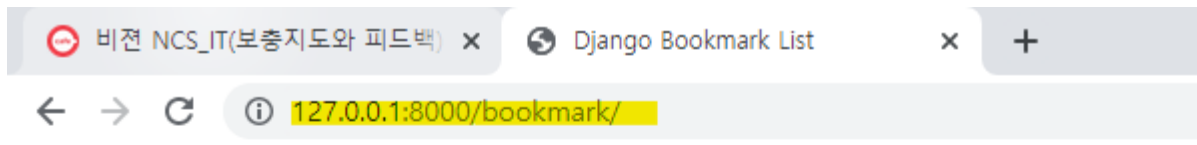
◆ 브라우저에 <http://127.0.0.1:8000/> 를 입력해서 들어갔을 때



◆ ① 관리자 모드를 클릭하면 주소창이 다음과 같이 바뀌면서 관리자모드 화면이 나옴



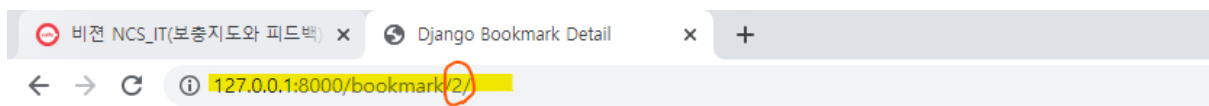
◆ ②북마크를 클릭해주면 주소창이 다음과 같이 바뀌면서 북마크 리스트들이 나온다



Bookmark List

- [우리들의 카페](#)
- [에스파!](#)
- [블랙맘바!!!!!!!!!!!!](#)
- [넥슬레v블!](#)

◆ 북마크 리스트에서 리스트 하나를 선택했을 때 id값이 넘어가면서 상세보기하기



에스파!

- https://www.smtown.com/search?keyword=%EC%97%90%EC%8A%A4%ED%8C%8C&star_idx=10766

☞ <참고>bookmark_detail.html 에서 9번째 라인에 `{{objects.title}}`을 쓰면 결과 달라짐!



- https://www.youtube.com/watch?v=HwVzLNx5_ms