

```
MySQL 5.6 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.6.27-log MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database jango
-> ;
Query OK, 1 row affected (0.34 sec)

mysql> use jango
Database changed
mysql> _
```

root/1234 jango

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'jango',
        'USER': 'root',
        'PASSWORD': '1234',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

pip install pymysql

pip install mysqlclient

```
명령 프롬프트
(venv1) G:\2019\python\file>pip install mysqlclient-1.3.13-cp36-cp36m-win_amd64.whl
mysqlclient-1.3.13-cp36-cp36m-win_amd64.whl is not a supported wheel on this platform.

(venv1) G:\2019\python\file>pip install mysqlclient-1.3.13-cp36-cp36m-win32.whl
Processing g:\2019\python\file\mysqlclient-1.3.13-cp36-cp36m-win32.whl
Installing collected packages: mysqlclient
Successfully installed mysqlclient-1.3.13

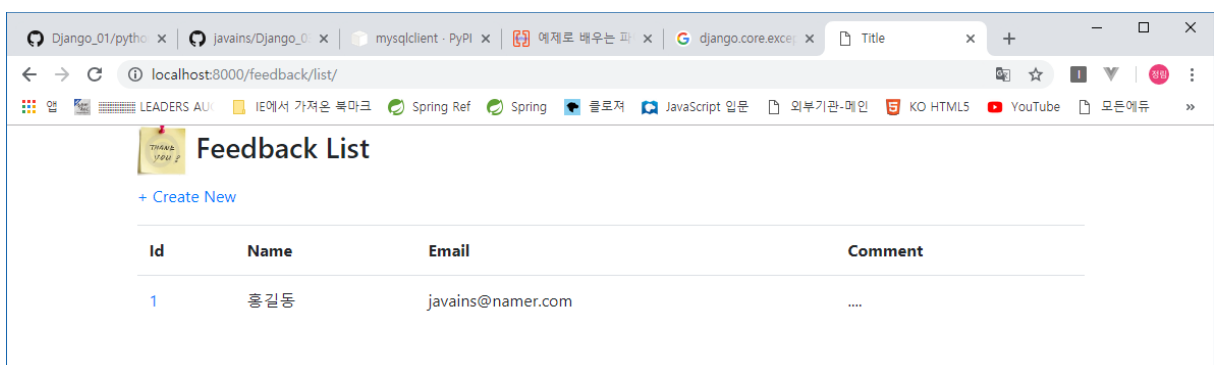
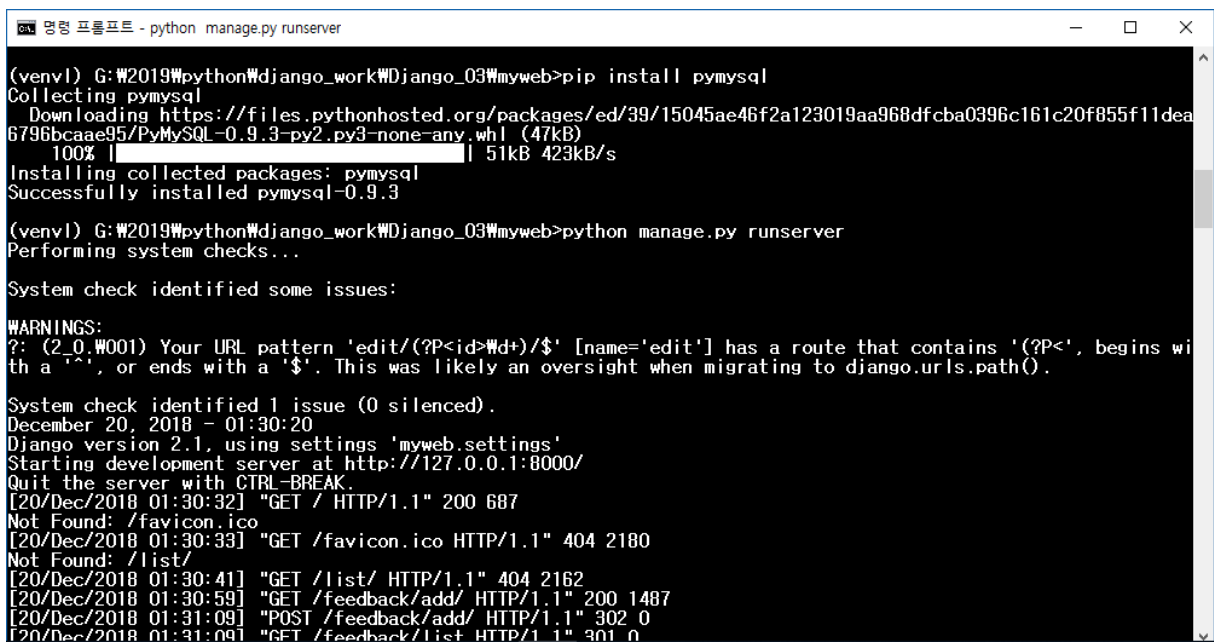
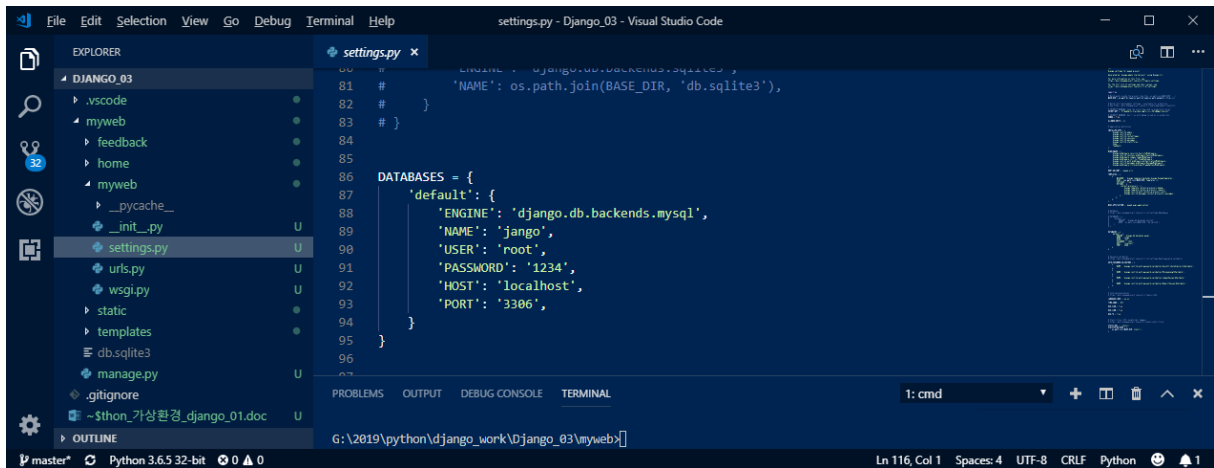
(venv1) G:\2019\python\file>
```

<https://www.lfd.uci.edu/~gohlke/pythonlibs/#mysqlclient>

설치가 잘 안되서 가상환경에 아래파일 다운받아서 실행해서 설치 성공

(venv1) G:\2019\python\django_work\Django_03\myweb>python manage.py makemigrations

(venv1) G:\2019\python\django_work\Django_03\myweb>python manage.py migrate

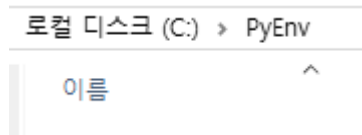


```
MySQL 5.6 Command Line Client
-> ;
Query OK, 1 row affected (0.34 sec)

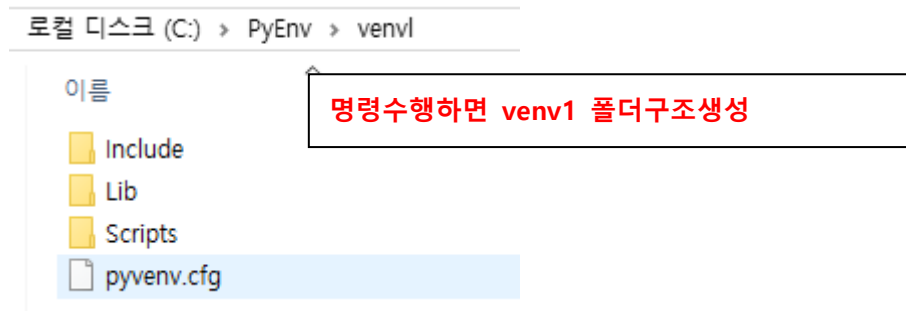
mysql> use jango
Database changed
mysql> show tables
-> ;
+-----+
| Tables_in_jango |
+-----+
| auth_group       |
| auth_group_permissions |
| auth_permission  |
| auth_user        |
| auth_user_groups |
| auth_user_user_permissions |
| django_admin_log |
| django_content_type |
| django_migrations |
| django_session   |
| feedback_feedback |
+-----+
11 rows in set (0.03 sec)

mysql> select * from feedback_feedback
-> ;
+----+-----+-----+-----+-----+
| id | name  | email          | comment | createDate |
+----+-----+-----+-----+-----+
| 1  | 홍길동 | javains@namer.com | ...    | 2018-12-19 16:31:09.775389 |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

1. 가상환경 만들 폴더 생성



2. C:\Users\USER\AppData\Local\Programs\Python\Python36-32>
python Tools/Scripts/pyenv.py c:/PyEnv/venv1



3. C:\PyEnv\venv1\Scripts>**activate**

윈도우즈에서 가상환경을 활성화(activate)

명령 프롬프트

```
(venv1) C:\PyEnv\venv1\Scripts>
```

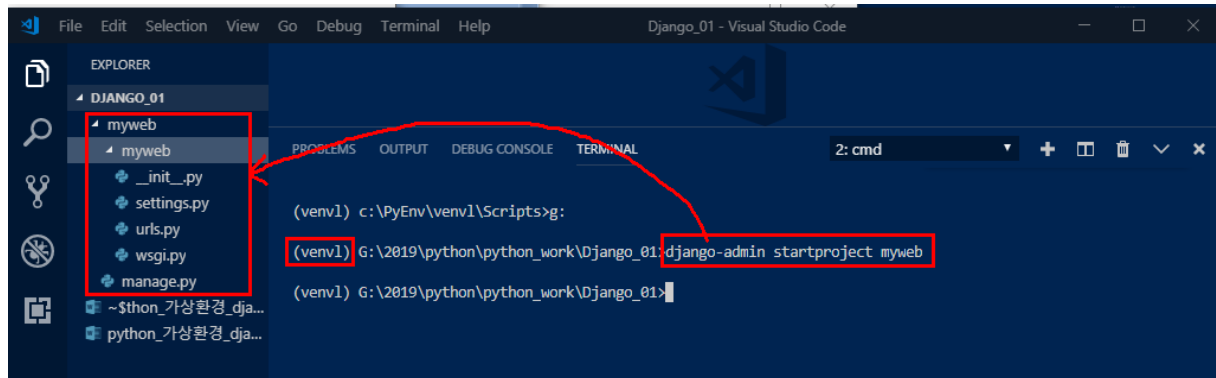
종료는 exit

4. (venv1) C:\PyEnv\venv1\Scripts>**python -m pip install --upgrade pip**
pip upgrade

5. (venv1) C:\PyEnv\venv1\Scripts>**pip install django==2.1**
django 2.1 버전 설치

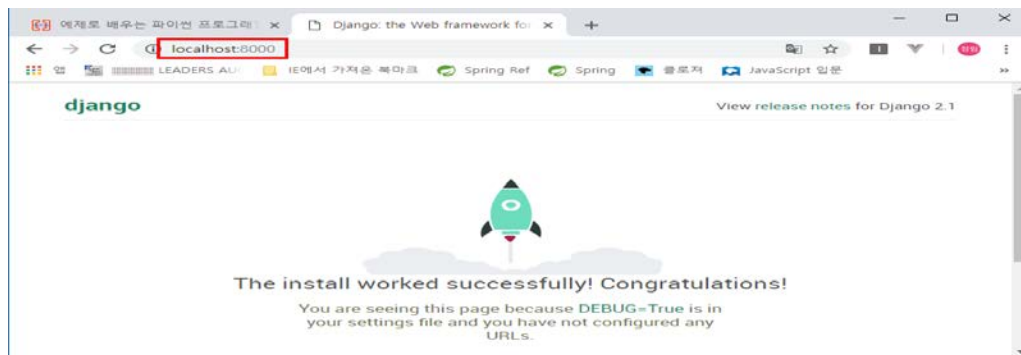
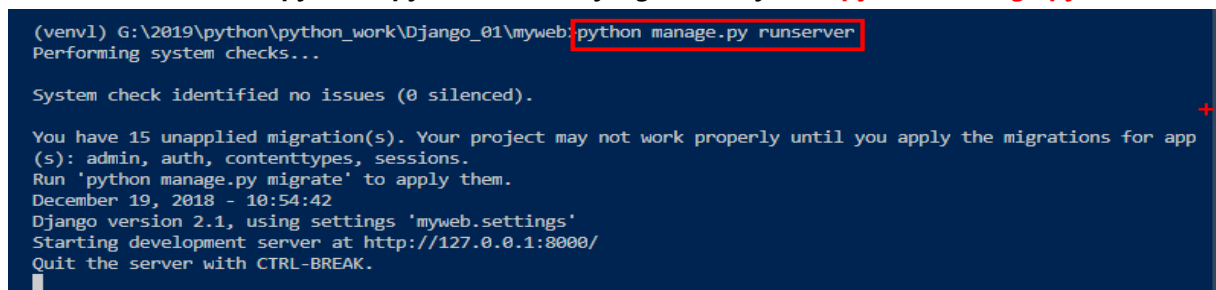
6. (venv1) G:\2019\python\python_work\DJango_01>**django-admin startproject myweb**

프로젝트를 만들 디렉토리로 이동한 후, 아래와 같이 "django-admin startproject 프로젝트명" 를 실행하여 새 프로젝트를 생성한다.

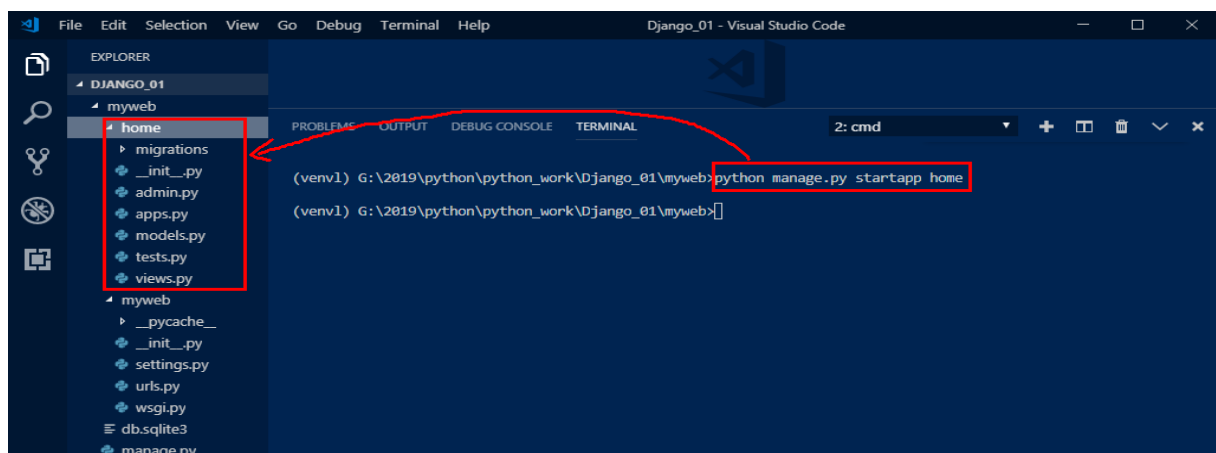


C:\WPyEnv\venv1\Scripts>**activate** 윈도우즈에서 가상환경을 활성화(activate)

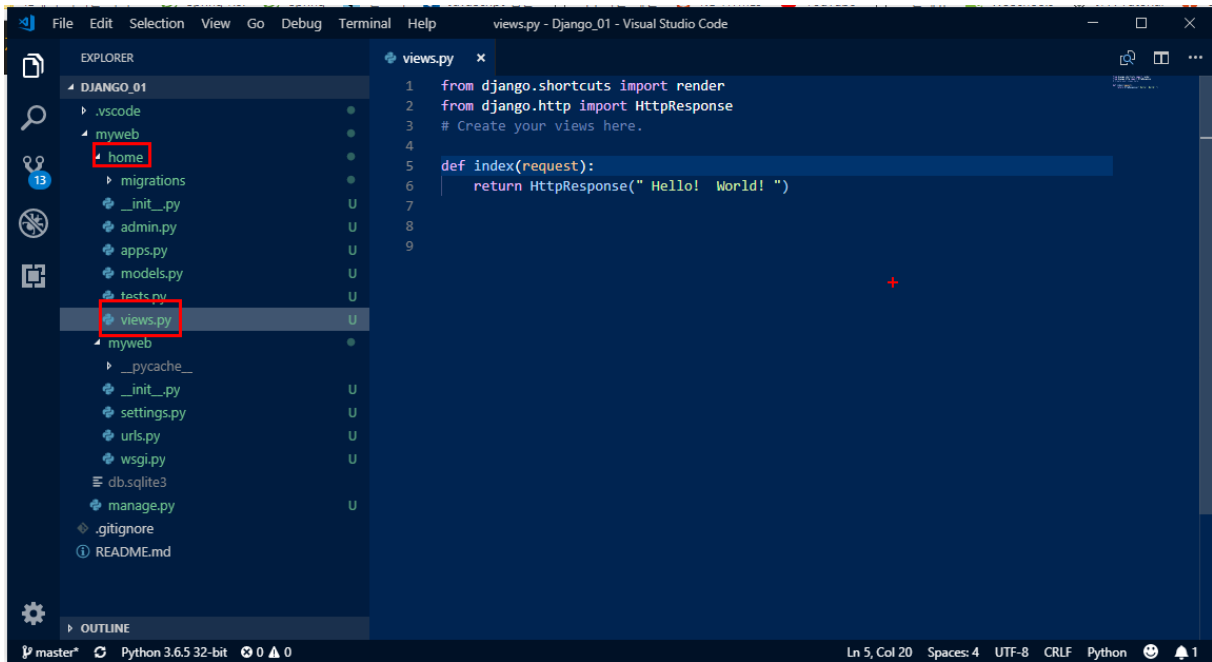
7. (venv1) G:\2019\python\python_work\DJango_01\myweb>**python manage.py runserver**



8. (venv1) G:\2019\python\python_work\DJango_01\myweb>**python manage.py startapp home**

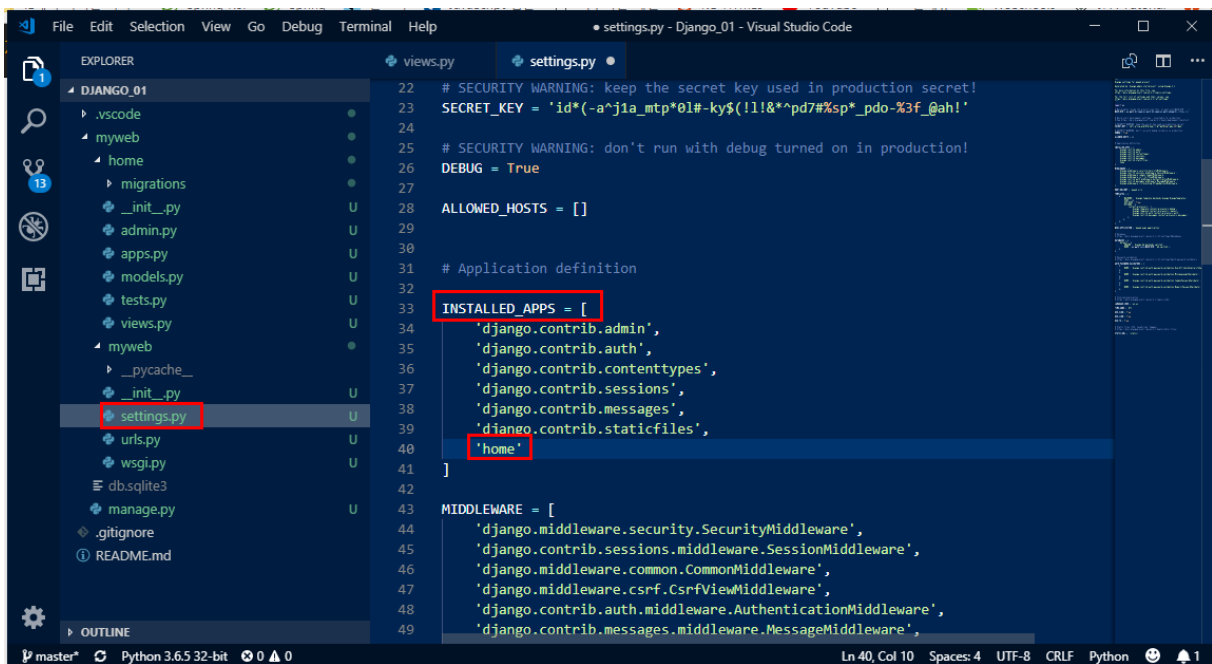


https://github.com/javains/Django_01.git 연동



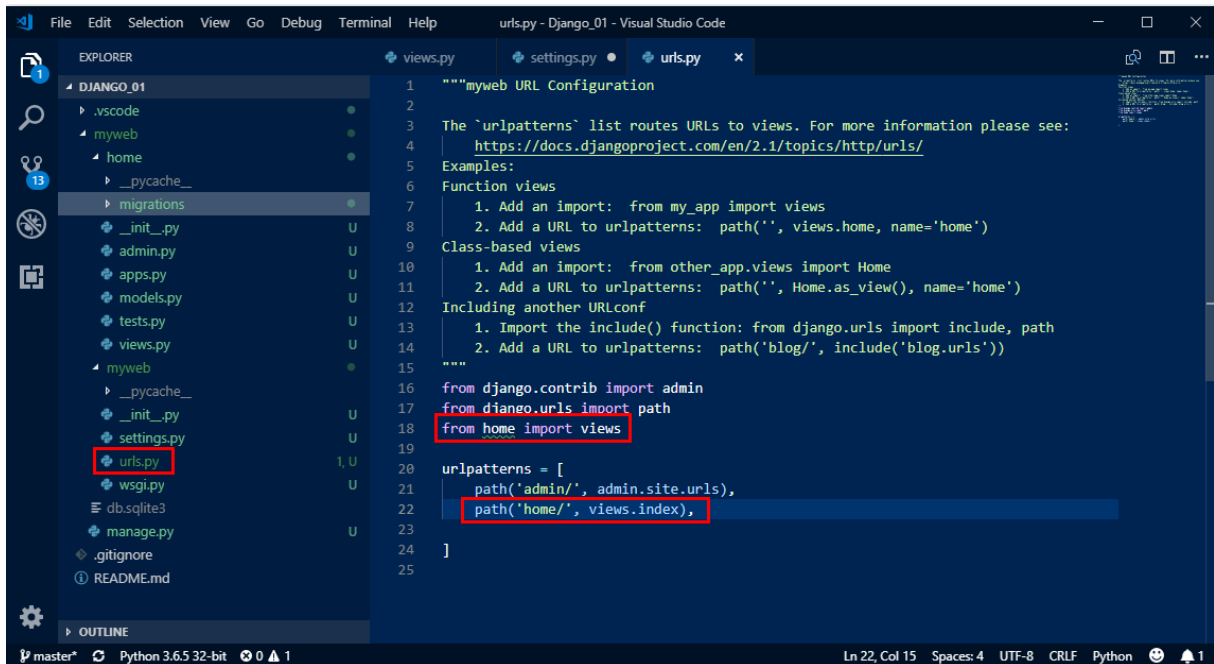
The screenshot shows the Visual Studio Code interface with the Django_01 project open. The Explorer panel on the left shows the project structure, with the 'home' directory and 'views.py' file highlighted. The main editor displays the contents of 'views.py'.

```
1 from django.shortcuts import render
2 from django.http import HttpResponse
3 # Create your views here.
4
5 def index(request):
6     return HttpResponse(" Hello! World! ")
7
8
9
```

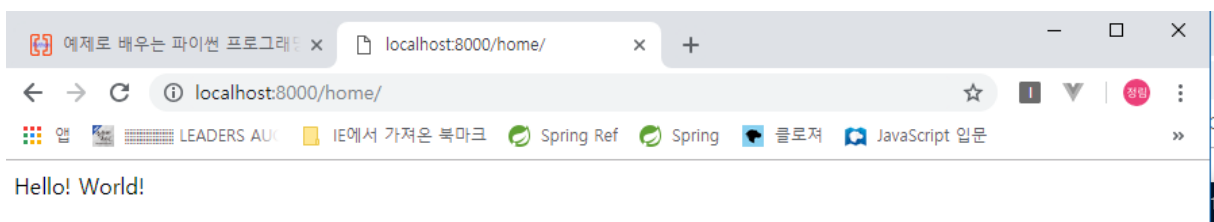


The screenshot shows the Visual Studio Code interface with the Django_01 project open. The Explorer panel on the left shows the project structure, with the 'settings.py' file highlighted. The main editor displays the contents of 'settings.py'.

```
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = 'id*(-a~j1a_mtp*0l#-ky$(!l!&*^pd7#%sp*_pdo-%3f_@ah!'
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30 # Application definition
31
32 INSTALLED_APPS = [
33     'django.contrib.admin',
34     'django.contrib.auth',
35     'django.contrib.contenttypes',
36     'django.contrib.sessions',
37     'django.contrib.messages',
38     'django.contrib.staticfiles',
39     'home'
40 ]
41
42 MIDDLEWARE = [
43     'django.middleware.security.SecurityMiddleware',
44     'django.contrib.sessions.middleware.SessionMiddleware',
45     'django.middleware.common.CommonMiddleware',
46     'django.middleware.csrf.CsrfViewMiddleware',
47     'django.contrib.auth.middleware.AuthenticationMiddleware',
48     'django.contrib.messages.middleware.MessageMiddleware',
49
```



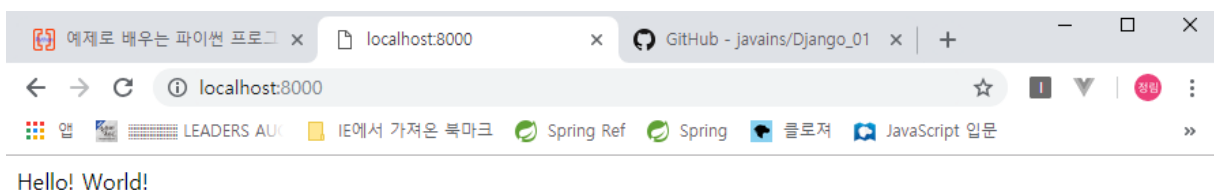
(venv) G:\2019Wpython\django_work\DJango_01\myweb>python manage.py runserver



```

urlpatterns = [
    path('admin/', admin.site.urls),
    path("", views.index),
]

```



템플릿 (Template)은 View로부터 전달된 데이터를 템플릿에 적용하여 Dynamic 한 웹페이지를 만드는데 사용된다.

Django 개발 가이드라인은 "App폴더/templates/템플릿파일" 처럼, 각 App 폴더 밑에 templates 서브폴더를 만들고 다시 그 안에 App명을 사용하여 서브폴더를 만든 후 템플릿 파일을 그 안에 넣기를 권장한다 (예: /home/templates/index.html).

The screenshot shows the Visual Studio Code interface with the Django project structure in the Explorer on the left. The project is named 'DJANGO_01' and contains a 'myweb' app. The 'views.py' file is selected in the Explorer and is also open in the editor. The code in 'views.py' is as follows:

```
1 from django.shortcuts import render
2 from django.http import HttpResponse
3 # Create your views here.
4
5 def index(request):
6     msg='hello django...'
7     #return HttpResponse(" Hello! World! ")
8     return render(request,'index.html',{'msg':msg})
9
10
11
```

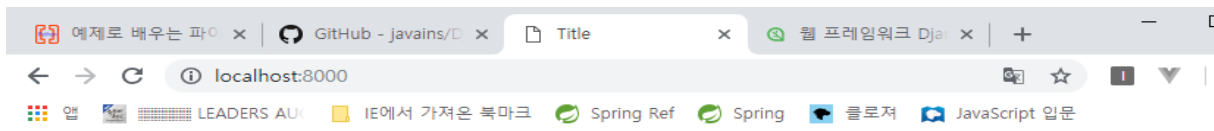
The screenshot shows the Visual Studio Code interface with the 'index.html' file selected in the Explorer and open in the editor. The code in 'index.html' is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8     <h1>index</h1>
9     <h1>{{msg}}</h1>
10 </body>
11 </html>
```

Below the editor, the terminal output is shown. The command 'python manage.py runserver' has been executed, and the output is:

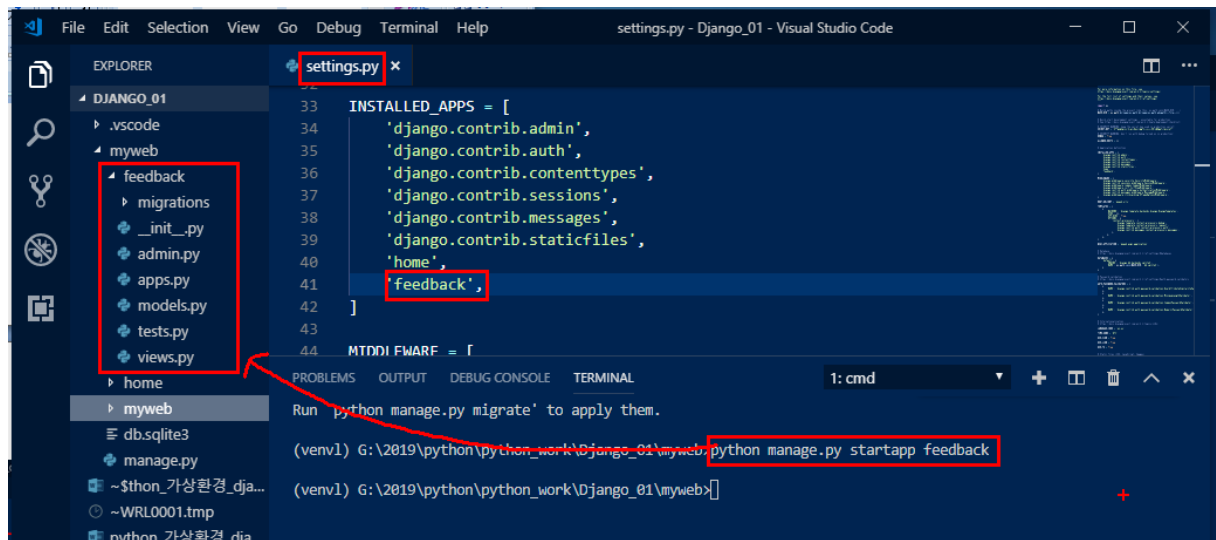
```
PS G:\2019\python\django_work\DJango_01\myweb> python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
```

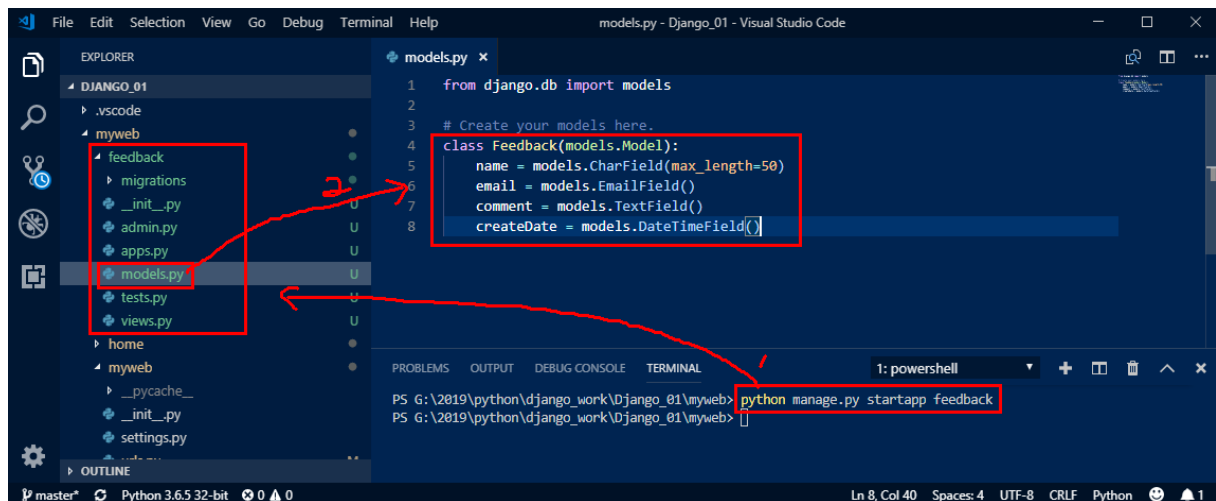


index

hello django...

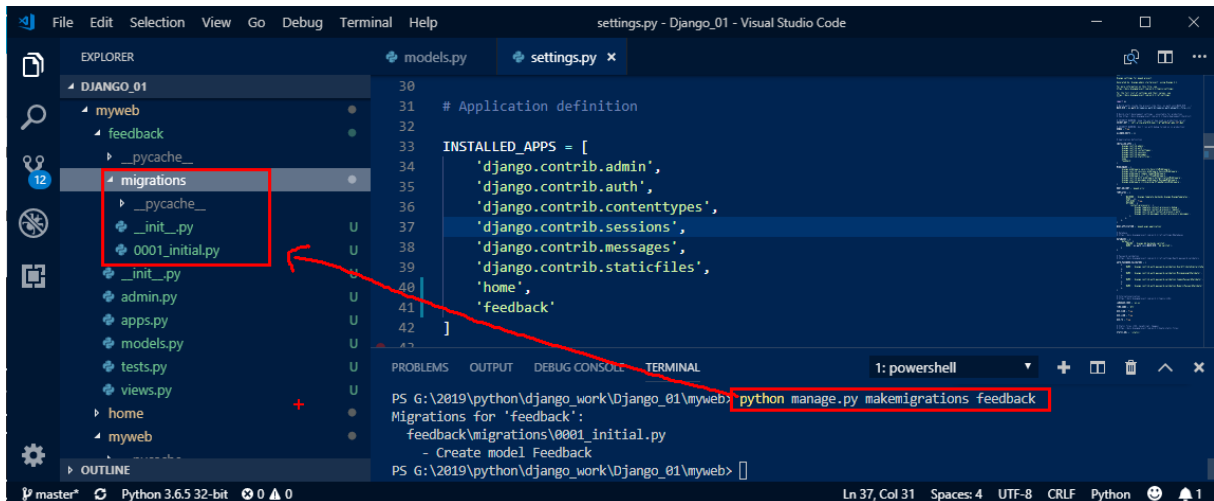


Django 모델 (Model)

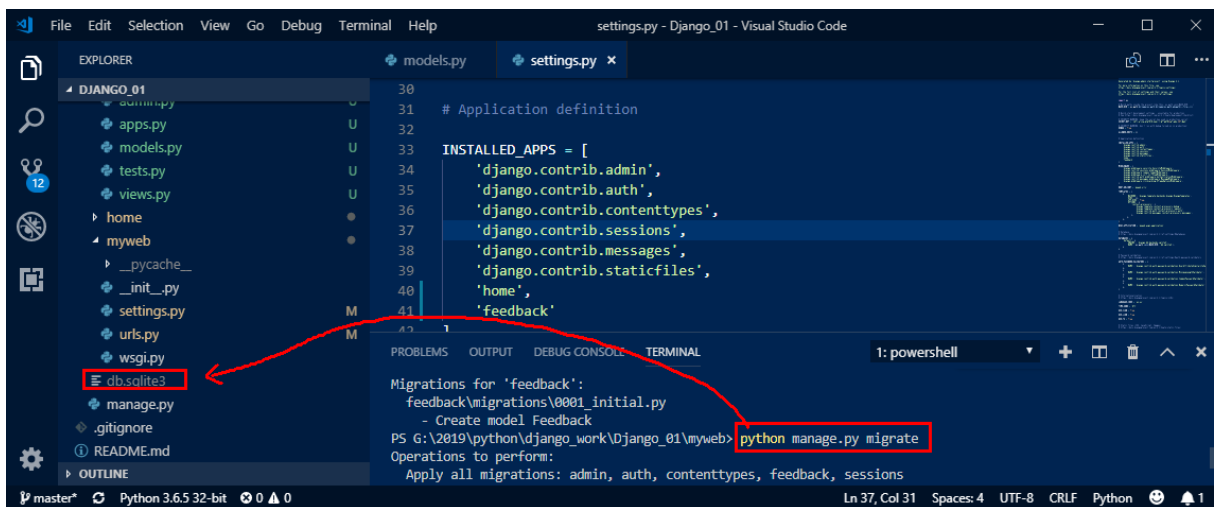


```
class Feedback(models.Model):
    name = models.CharField(max_length=50)
    email = models.EmailField()
    comment = models.TextField(null=True)
    createDate = models.DateTimeField(auto_now_add=True)
```

settings.py => INSTALLED_APPS 리스트에 'feedback'

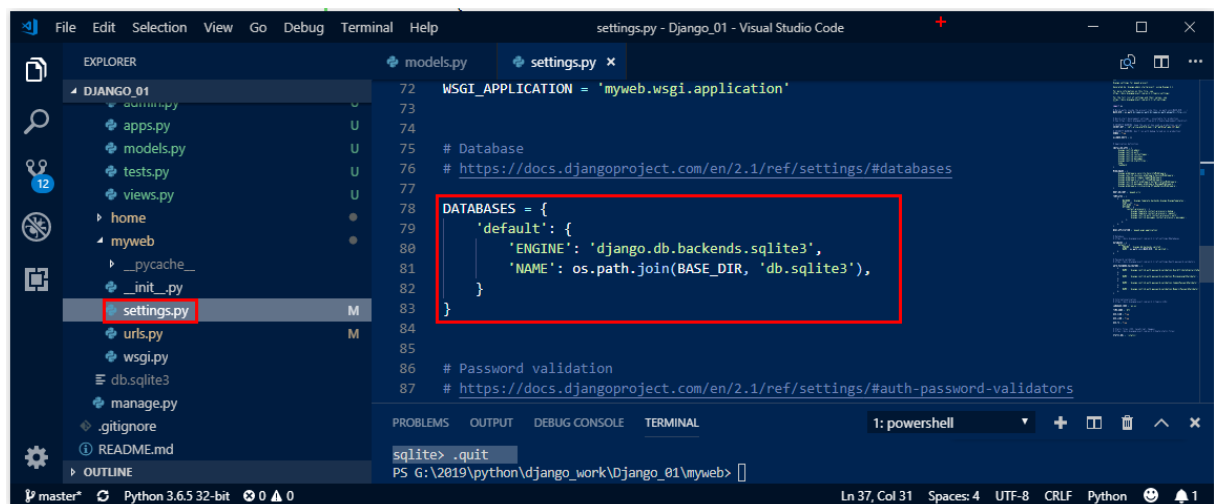


1. PS G:\W2019\python\django_work\Wdjango_01\Wmyweb> **python manage.py makemigrations feedback**
==> feedback app 에 model 변화가 있는지 확인만한다.
2. PS G:\W2019\python\django_work\Wdjango_01\Wmyweb> **python manage.py migrate**
==> model을 DataBase에 적용한다.



G:\W2019\python\django_work\Wdjango_01\Wmyweb> **python manage.py dbshell**
 sqlite> **.tables**
 sqlite> **pragma table_info(feedback_feedback);**
 sqlite> **.quit**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: python
PS G:\2019\python\django_work\Django_01\myweb> python manage.py dbshell
SQLite version 3.26.0 2018-12-01 12:34:55
Enter ".help" for usage hints.
sqlite> .tables
auth_group          django_admin_log
auth_group_permissions  django_content_type
auth_permission     django_migrations
auth_user           django_session
auth_user_groups    feedback_feedback
auth_user_user_permissions
sqlite> pragma table_info(feedback_feedback);
0|id|integer|1||1
1|name|varchar(50)|1||0
2|email|varchar(254)|1||0
3|comment|text|1||0
4|createDate|datetime|1||0
sqlite> .quit
```



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'MyDB',
        'USER': 'user1',
        'PASSWORD': 'pwd',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

- django.db.backends.postgresql
- django.db.backends.mysql
- django.db.backends.sqlite3
- django.db.backends.oracle

```
from feedback.models import *
from datetime import datetime
```

INSERT

```
# Feedback 객체 생성
fb = Feedback(name = 'Kim', email = 'kim@test.com', comment='Hi',
createDate=datetime.now())
fb.save()    # 새 객체 INSERT
```

SELECT

```
for f in Feedback.objects.all():
    s += str(f.id) + ' : ' + f.name + '\n'

row = Feedback.objects.get(pk=1)
print(row.name)

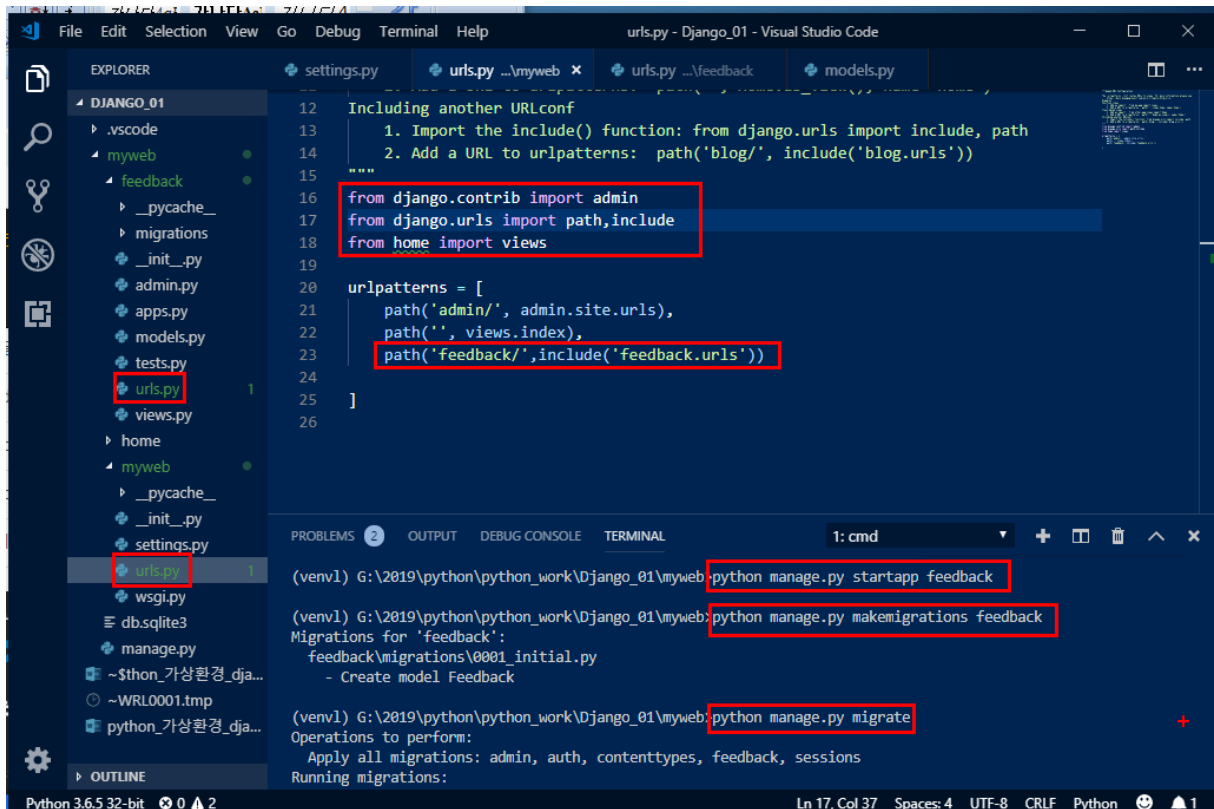
rows = Feedback.objects.filter(name='Kim')
n = Feedback.objects.count()    # 데이터의 갯수(row 수)
rows = Feedback.objects.distinct('name')
rows = Feedback.objects.order_by('name').first()
rows = Feedback.objects.order_by('name').last()
```

UPDATE

```
fb = Feedback.objects.get(pk=1)
fb.name = 'Park'
fb.save()
```

DELETE

```
fb = Feedback.objects.get(pk=2)
fb.delete()
```



```

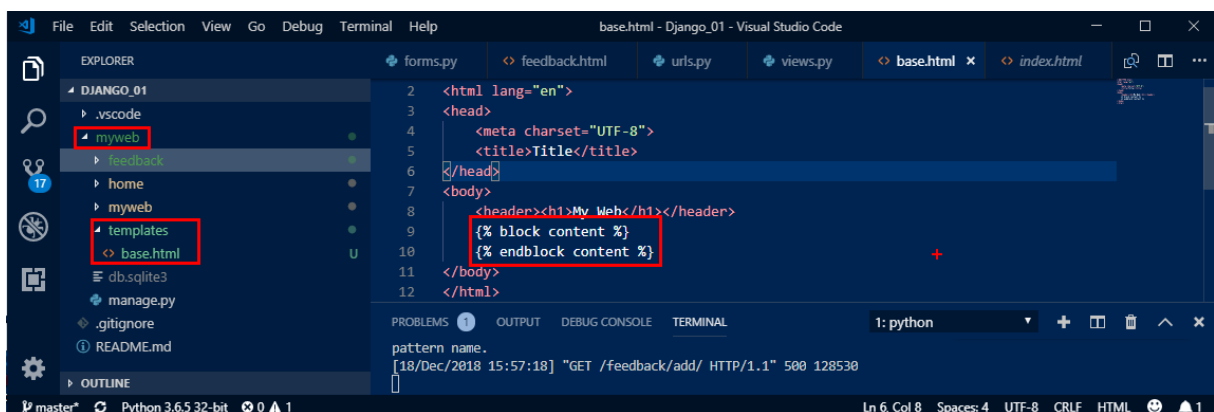
from django.contrib import admin
from django.urls import path
from feedback import views
urlpatterns = [
    path('add/', views.add),
]

```

```

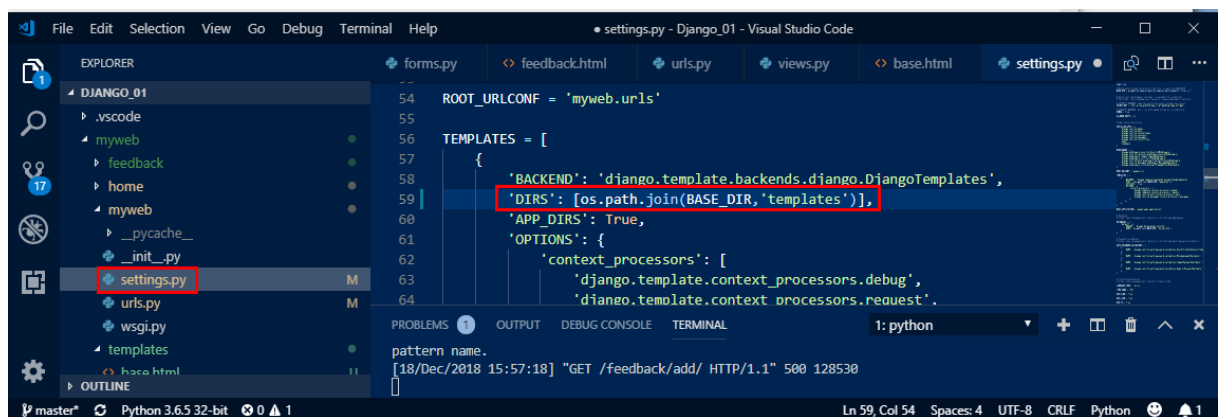
urlpatterns = [
    path('add/', views.add),
    path('update/', views.update),
    path('list/', views.list),
]

```

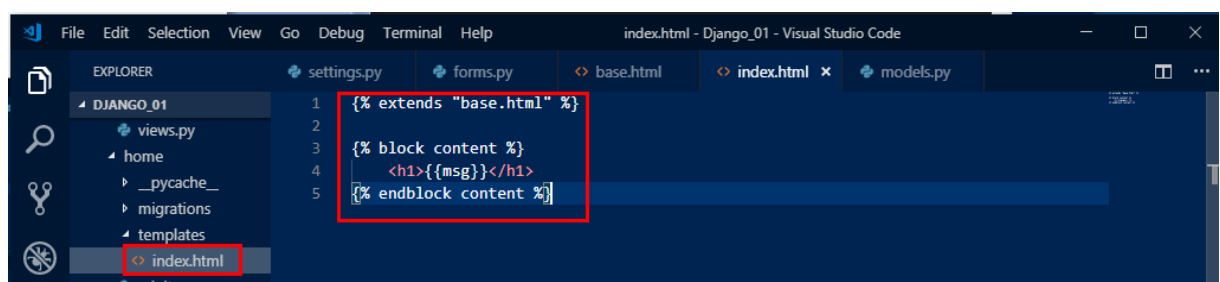


`./templates/base.html` 이라는 **Base** 템플릿을 만들었는데,
이 파일 안에 각 웹페이지에서 변경 혹은 삽입할 영역을 `{% block 블럭명 %}` 으로 지정한다. 여기서
블럭명을 **content**로 정하여 `{% block content %}` 으로 표시

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <header>...My Web...</header>
  {% block content %}
  {% endblock content %}
</body>
</html>
```



템플릿 위치 셋팅



...My Web...
Hello Django...

The screenshot shows the Visual Studio Code interface with the Django project 'DJANGO_01' open. The Explorer panel on the left shows the project structure, with the 'feedback' app and 'forms.py' file highlighted. The main editor displays the content of 'forms.py':

```

1 from django.forms import ModelForm
2 from .models import Feedback
3
4 class FeedbackForm(ModelForm):
5     class Meta:
6         model = Feedback
7         fields = ['id', 'name', 'email', 'comment']

```

The bottom terminal panel shows the command 'Applying feedback.0002_auto_20181218_1531... OK' and the file path 'PS G:\2019\python\django_work\DJango_01\myweb>'.

The screenshot shows the Visual Studio Code interface with the Django project 'DJANGO_01' open. The Explorer panel on the left shows the project structure, with the 'feedback' app and 'forms.py' file highlighted. The main editor displays the content of 'forms.py':

```

1 from django.forms import ModelForm
2 from .models import Feedback
3
4 class FeedbackForm(ModelForm):
5     class Meta:
6         model = Feedback
7         fields = ['name', 'email', 'comment']

```

The bottom terminal panel shows the command 'Applying feedback.0002_auto_20181218_1531... OK' and the file path 'PS G:\2019\python\django_work\DJango_01\myweb>'.

CSRF (Cross Site Request Forgeries)는 웹 해킹 기법의 하나로 Django는 이를 방지하기 위한 기능을 기본적으로 제공하고 있다. Django에서 HTTP POST, PUT, DELETE을 할 경우 이 태그를 넣어 주어야 한다

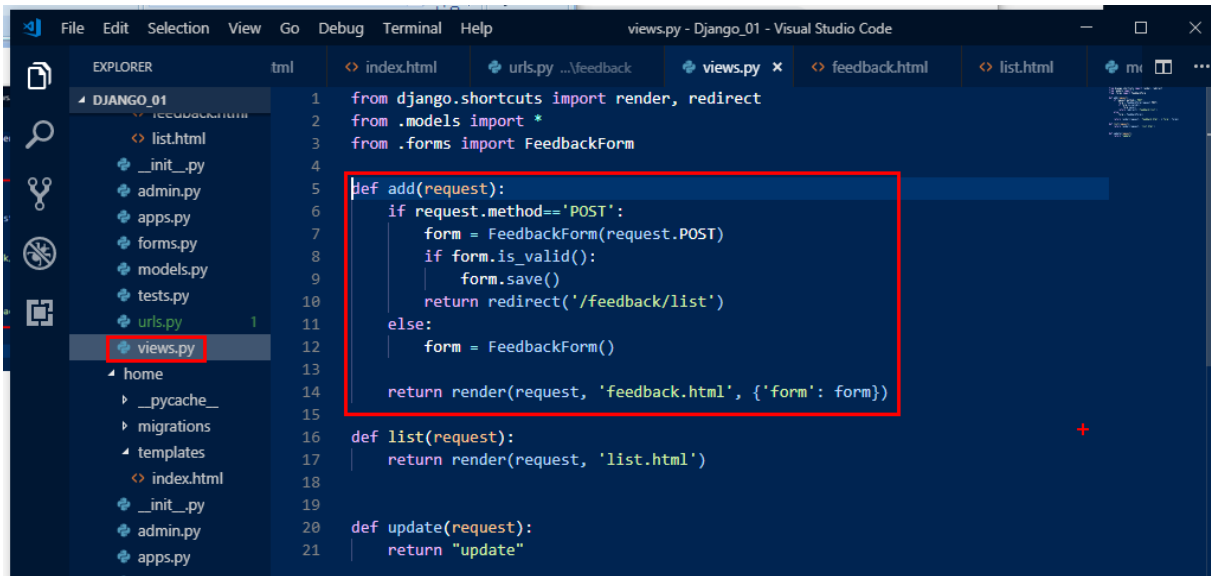
The screenshot shows the Visual Studio Code interface with the Django project 'DJANGO_01' open. The Explorer panel on the left shows the project structure, with the 'feedback' app and 'feedback.html' file highlighted. The main editor displays the content of 'feedback.html':

```

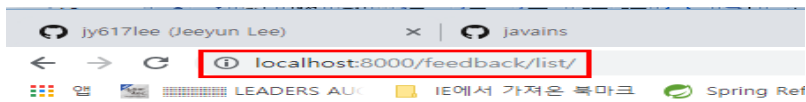
1 {% extends "base.html" %}
2
3 {% block content %}
4     <p>
5         <a href="{% url 'list' %}">Goto Feedback List</a>
6     </p>
7
8     <div>
9         <form method="POST">
10             {% csrf_token %}
11             {{ form.as_p }}
12             <button type="submit">저장</button>
13         </form>
14     </div>
15
16 {% endblock content %}

```

The bottom terminal panel shows the command 'Quit the server with CTRL-BREAK.' and the file path 'PS G:\2019\python\django_work\DJango_01\myweb>'.

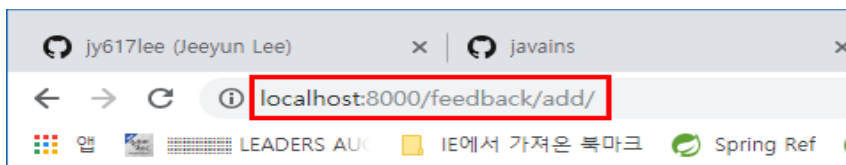


```
1 from django.shortcuts import render, redirect
2 from .models import *
3 from .forms import FeedbackForm
4
5 def add(request):
6     if request.method == 'POST':
7         form = FeedbackForm(request.POST)
8         if form.is_valid():
9             form.save()
10            return redirect('/feedback/list')
11        else:
12            form = FeedbackForm()
13
14    return render(request, 'feedback.html', {'form': form})
15
16 def list(request):
17     return render(request, 'list.html')
18
19 def update(request):
20     return "update"
```



...My Web...

list



...My Web...

[Goto Feedback List](#)

Name:

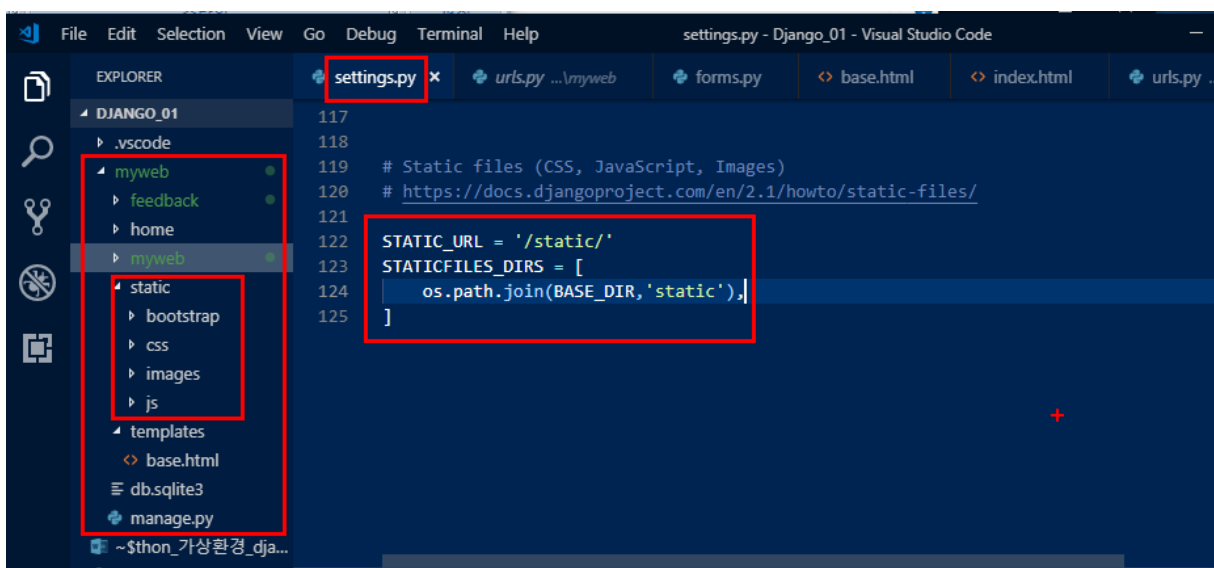
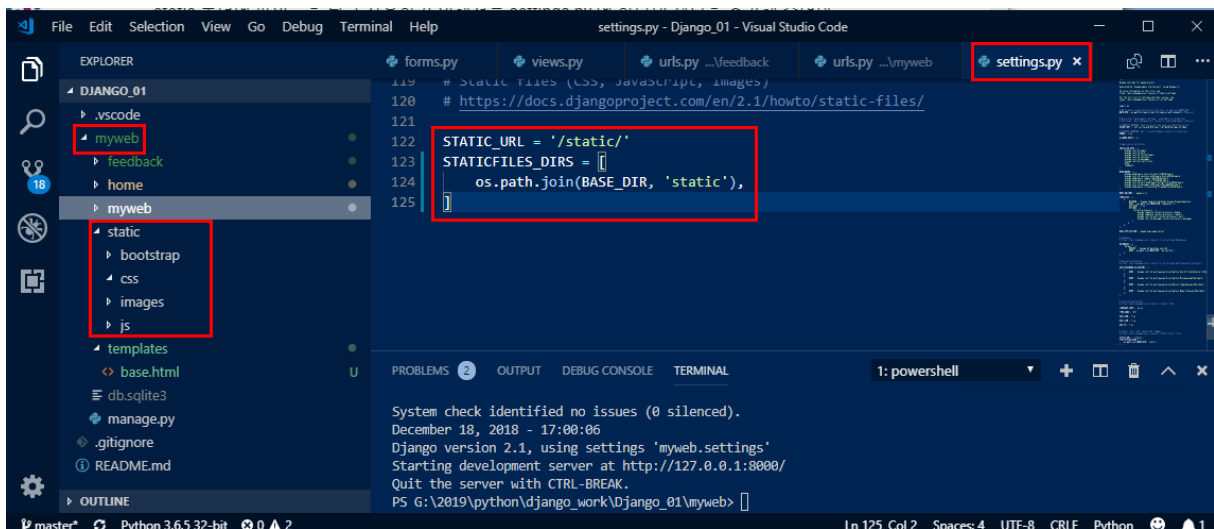
Email:

Comment:


```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL 1: cmd

(env1) G:\2019\python\python_work\Django_01\myweb>python manage.py dbshell
SQLite version 3.26.0 2018-12-01 12:34:55
Enter ".help" for usage hints.
sqlite> .tables
auth_group          django_admin_log
auth_group_permissions django_content_type
auth_permission      django_migrations
auth_user            django_session
auth_user_groups     feedback_feedback
auth_user_user_permissions
sqlite> select * from feedback_feedback
...>
1|홍길동|javains@namer.com|... django test...|2018-12-19 03:10:20.166695
2|고길동|javains@namer.com|sample test...|2018-12-19 03:13:41.908771
sqlite> .quit

(env1) G:\2019\python\python_work\Django_01\myweb>
```



```

from django.contrib import admin
from django.urls import path
from feedback import views

urlpatterns = [
    path('add/', views.add,name='add'),
    path('list/', views.list,name='list'),
    path('edit/(?P<id>\d+)/$', views.edit,name='edit'),
]

```

```

from django.shortcuts import render, redirect
from .models import *
from .forms import FeedbackForm

def add(request):
    if request.method=='POST':
        form = FeedbackForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('/feedback/list')
    else:
        form = FeedbackForm()

    return render(request, 'feedback.html', {'form': form})

def list(request):
    #return render(request, 'list.html')
    feedbacks = Feedback.objects.all()
    return render(request, 'list.html', {'feedbacks': feedbacks})

def edit(request,id):
    fb = Feedback.objects.get(pk=id)
    if request.method=='POST':
        form = FeedbackForm(request.POST, instance=fb)
        if form.is_valid():
            form.save()
            return redirect('/feedback/list')
    else:
        form = FeedbackForm(instance=fb)

    return render(request, 'feedback.html', {'form': form})

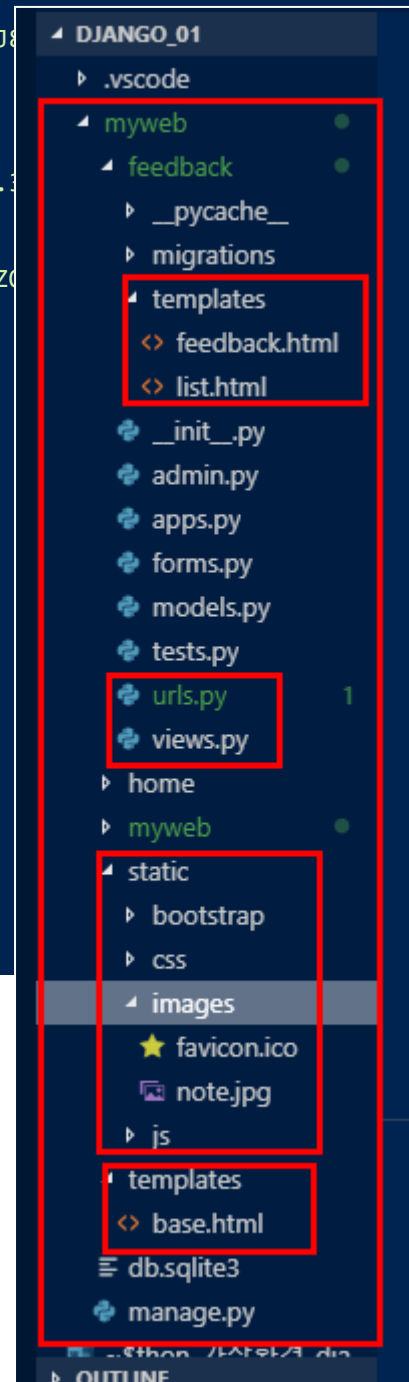
```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8
crossorigin="anonymous">
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
integrity="sha384-
ChfqqxUzUCnJK3+MxMPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZ
crossorigin="anonymous"></script>
</head>
<body>
    <div class="container">
        {% block title %}
        {% endblock title %}
    </div>

    <div class="container">
        {% block content %}
        {% endblock content %}
    </div>
</body>
</html>

```



```

{% extends "base.html" %}

{% block title %}
    <h3>Your Feedback..</h3>
{% endblock title %}

{% block content %}
    <p>
        <a href="{% url 'list' %}">Goto Feedback List</a>
    </p>

    <div>
        <form method="POST">
            {% csrf_token %}
            <table class="table">
                {{ form.as_table }}
            </table>
            <button type="submit">저장</button>
        </form>
    </div>

{% endblock content %}

```

← → ↻ localhost:8000/feedback/add/

IE에서 가져온 북마크

Your Feedback..

[Goto Feedback List](#)

Name:

Email:

Comment:

저장

```

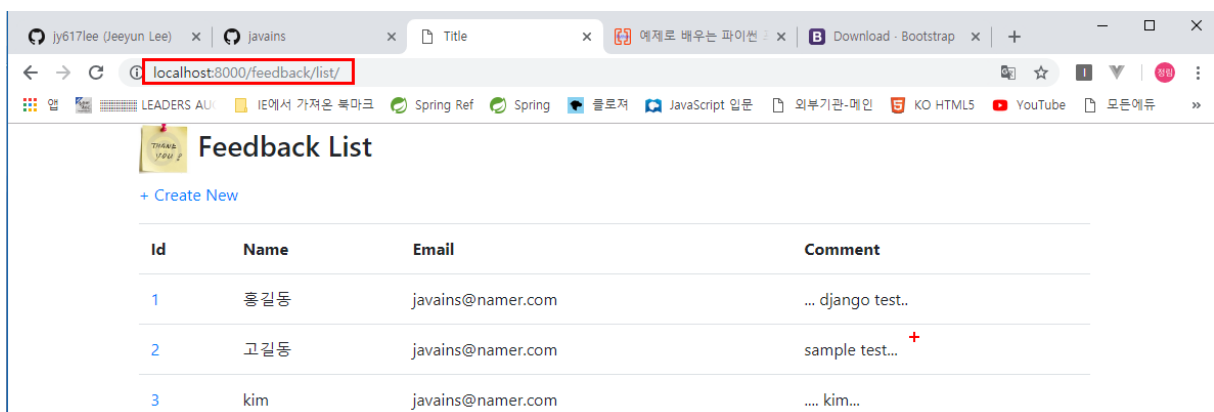
{% extends "base.html" %}
{% load staticfiles %}

{% block title %}
    <h3>
        
        Feedback List
    </h3>
{% endblock title %}

{% block content %}
    <p>
        <a href="{% url 'add' %}">+ Create New</a>
    </p>

    <table class="table">
        <tr><th>Id</th><th>Name</th><th>Email</th><th>Comment</th></tr>
        {% for item in feedbacks %}
            <tr>
                <td>
                    <a href="{% url 'edit' id=item.id %}">{{ item.id }}</a>
                </td>
                <td>{{ item.name }}</td>
                <td>{{ item.email }}</td>
                <td>{{ item.comment }}</td>
            </tr>
        {% endfor %}
    </table>
{% endblock content %}

```



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/feedback/list/'. The page content includes a title 'Feedback List' and a '+ Create New' link. Below this is a table with the following data:

Id	Name	Email	Comment
1	홍길동	javains@namer.com	... django test..
2	고길동	javains@namer.com	sample test... +
3	kim	javains@namer.com kim...

