

## [Django] #6 - Django 를 통한 사용자 등록 구현하기 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-10-17 오후 11:45

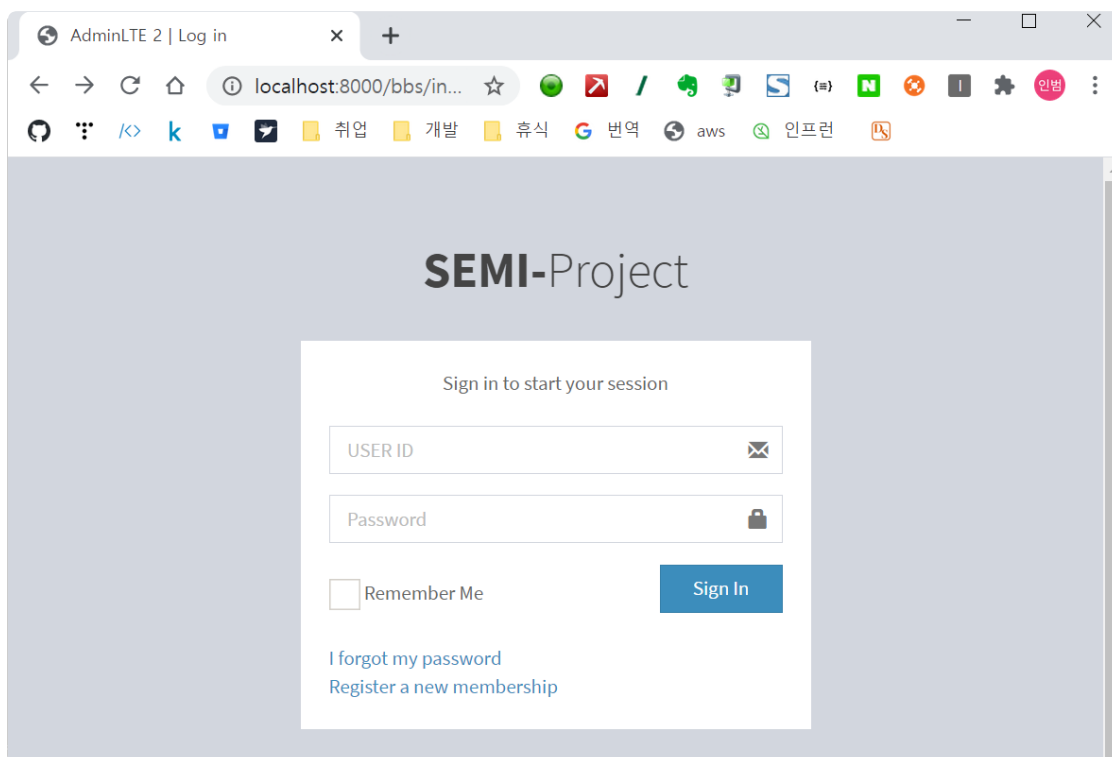
URL: <https://continuous-development.tistory.com/95?category=805760>

Django

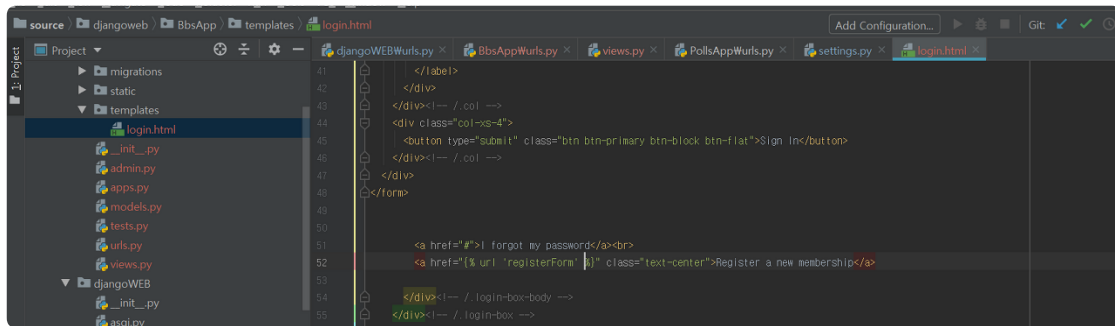
# [Django] #6 - Django 를 통한 사용자 등록 구현하기

2020. 9. 22. 11:08 수정 삭제 공개

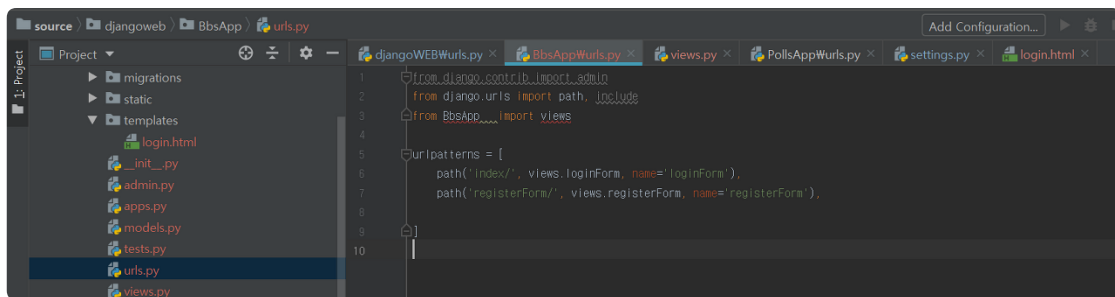
## Register a new membership



이 Register a new membership 에 대한 로직을 처리해 보자



해당 부분에 대해 href 를 수정하고 수정 된 부분에 맞게 url을 설정해준다.



path('registerForm/', views.registerForm, name='registerForm'),

1.

2.

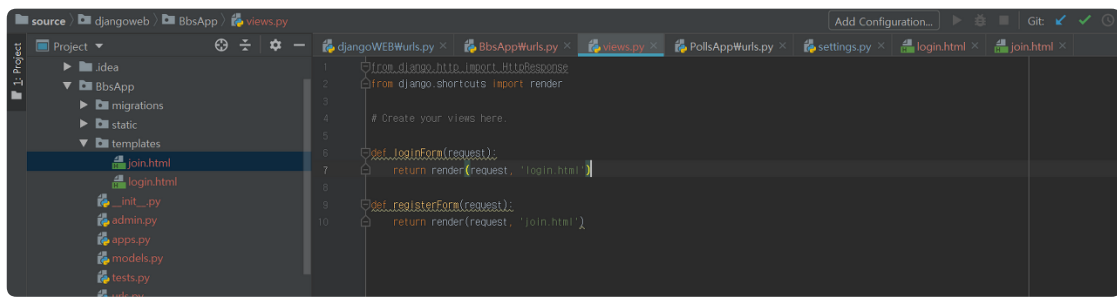
3.

url로 타고 들어오는 부분

view에 있는 함수

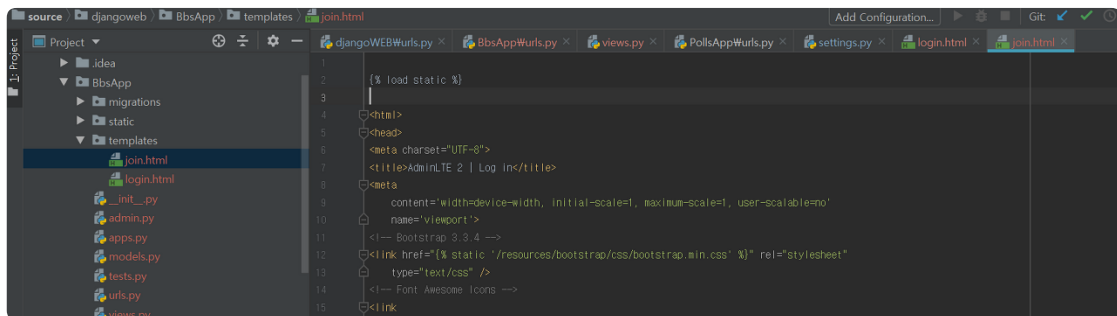
개발자가 인식할 수 있는 URL 부분

이다.

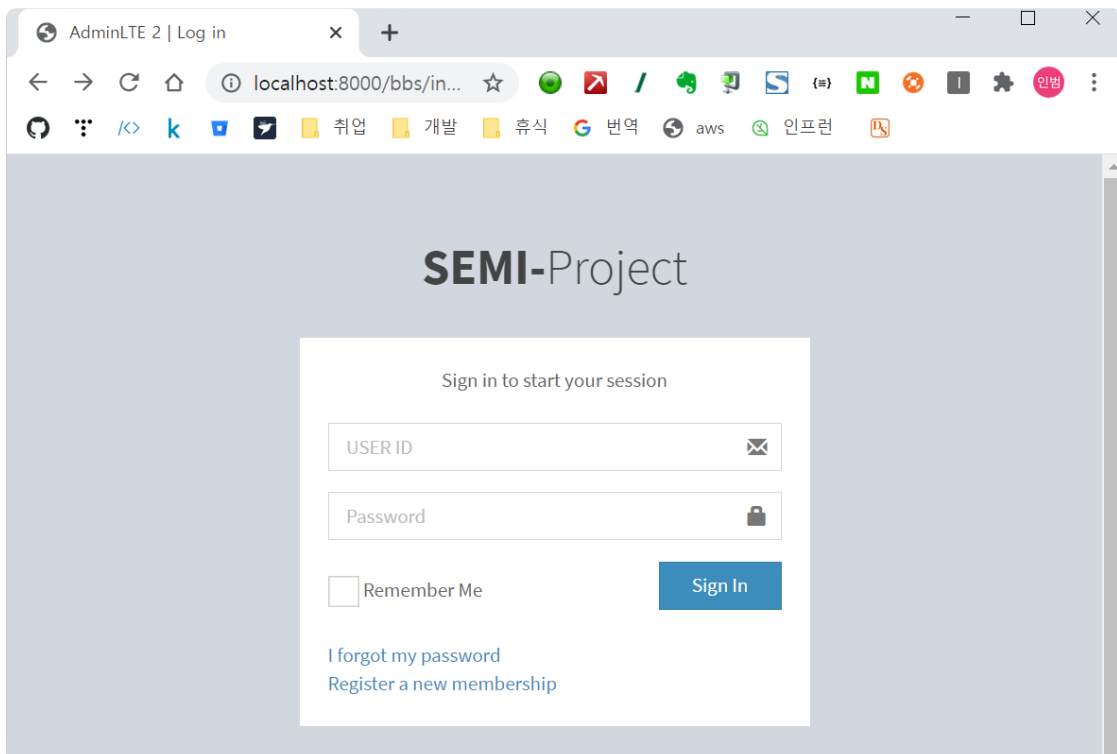


해당 하는 view에 해당 함수를 설정하고

join template 을 넣어준다.



이 후 사이트로 다시 가서 Register를 클릭하면





```

1 from django.contrib import admin
2 from django.urls import path, include
3 from BbsApp import views
4
5 urlpatterns = [
6     path('index/', views.loginForm, name='loginForm'),
7     path('registerForm/', views.registerForm, name='registerForm'),
8     path('register/', views.register, name='register'),
9 ]

```

모두 regist로 맞춰준다. 그다음 view도 설정을 해준다.

```

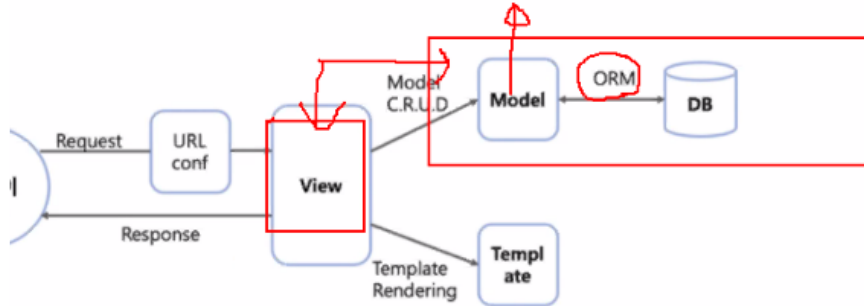
1 from django.http import HttpResponseRedirect
2 from django.shortcuts import render
3
4 # Create your views here.
5
6 def loginForm(request):
7     return render(request, 'login.html')
8
9 def registerForm(request):
10    return render(request, 'join.html')
11
12 def register(request):
13    return render(request, 'join.html')

```

우리는 지금 model을 통해 객체를 생성해야 한다.

밑에 작업이 필요하다.

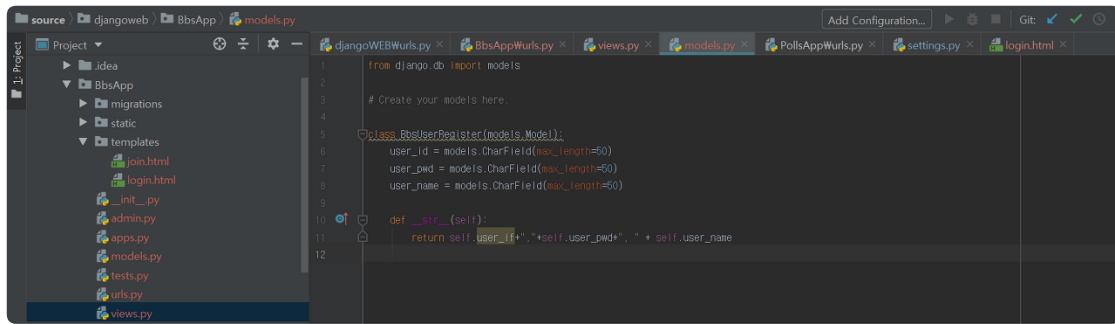
여기서는 View를 Template, Controller는 View라고 표현하며, MVC를 MVT 패턴이라고 한다. 모델은 데이터 베이스에 저장  
 | 의미하는 것이고, 템플릿은 사용자에게 보여주는 UI부분을, 뷰는 앞부분으로 데이터가 들어와서 데이터를 가져오고 적  
 | 결과를 템플릿에 전달하는 역할을 수행한다.



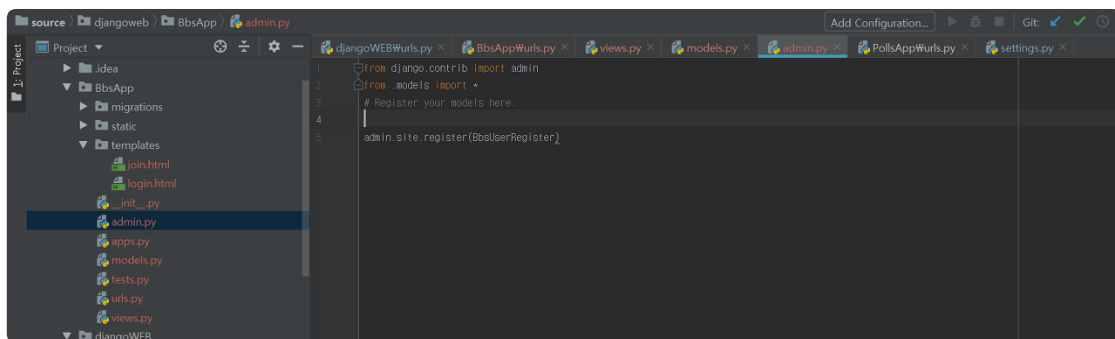
장고의 MVT 패턴

| 요청을 받고, 장고에서 MVT 패턴에 따라 처리하는 과정을 요약하면 다음과 같다.

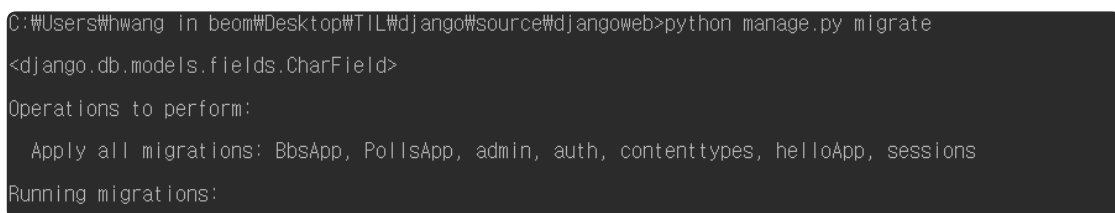
지금 이 작업이 필요하다.



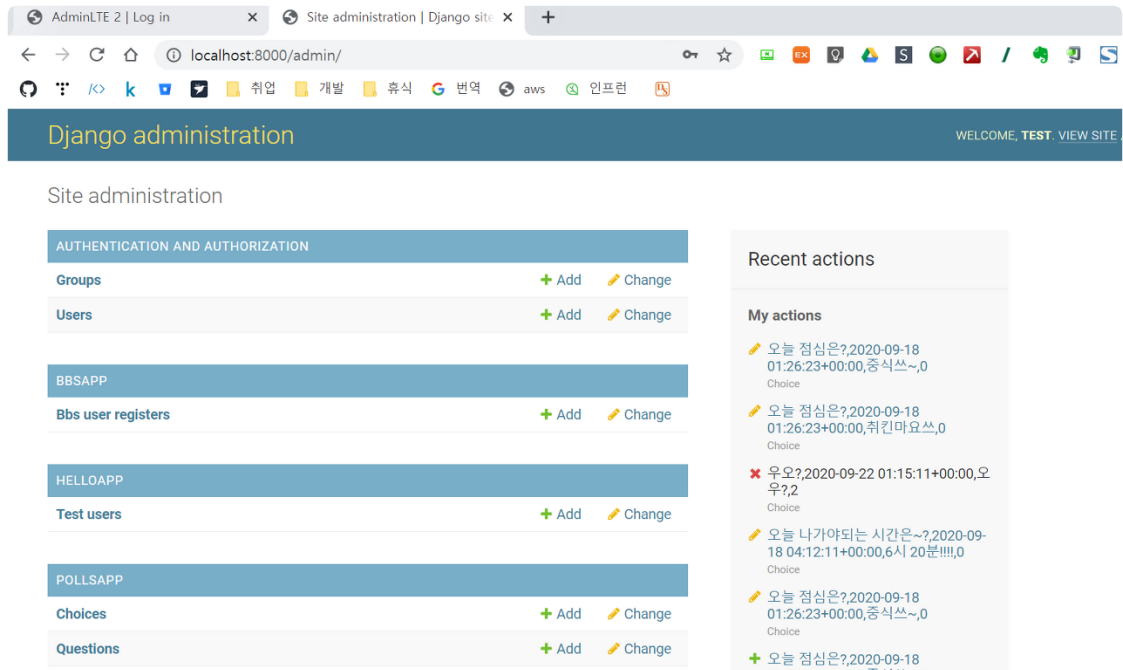
이렇게 모델 부분에 class 를 만들어준다. 그 뒤 admin에 해당하는 클래스를 사용한다고 등록해줘야 한다.



model을 만들거나 수정한 뒤에는 migrations를 해줘야한다.

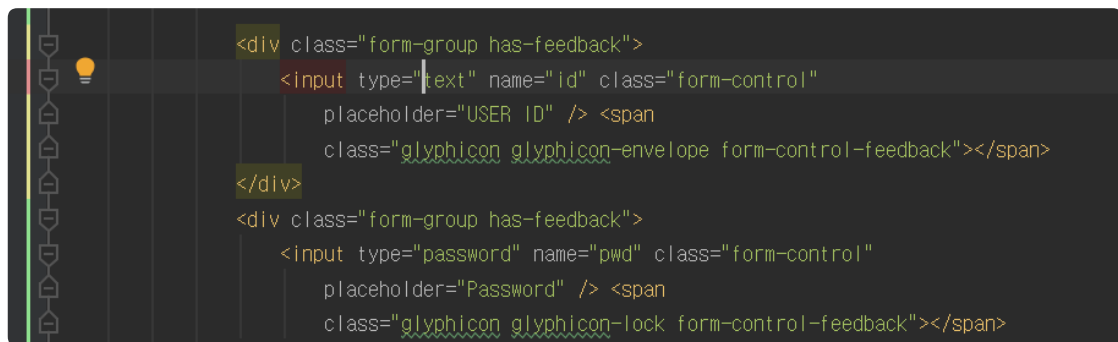


여기까지 진행한 후 admin 으로 접속하면

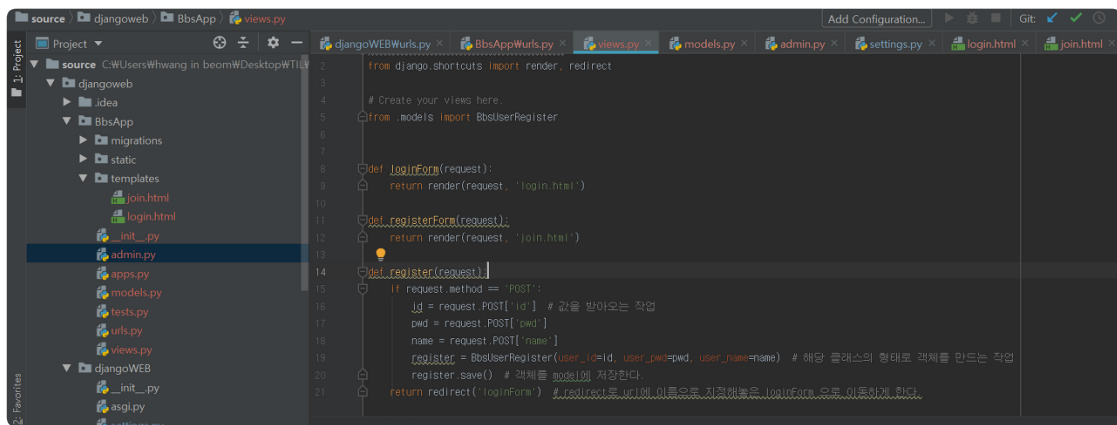


BBSAPP 부분에 해당 테이블이 생긴걸 볼 수 있다.

이제 다시 view 부분을 제대로 구현해보자.

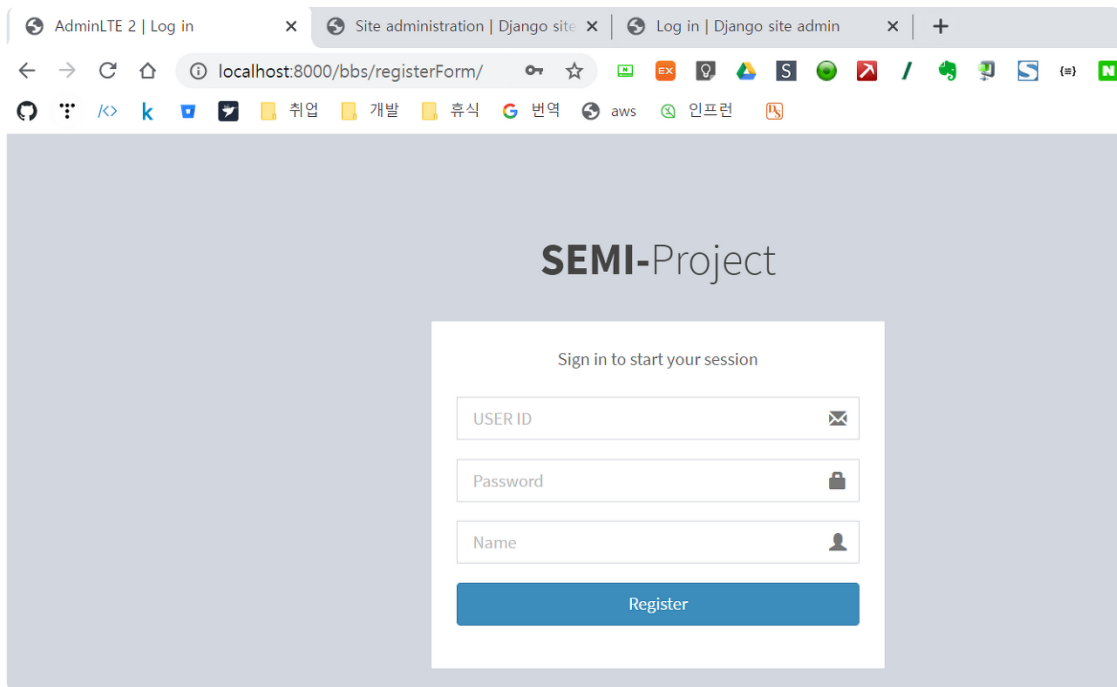


join 부분에서 name 변수에 값들을 담아 던진다. 이 값을 view 에서 받는다.



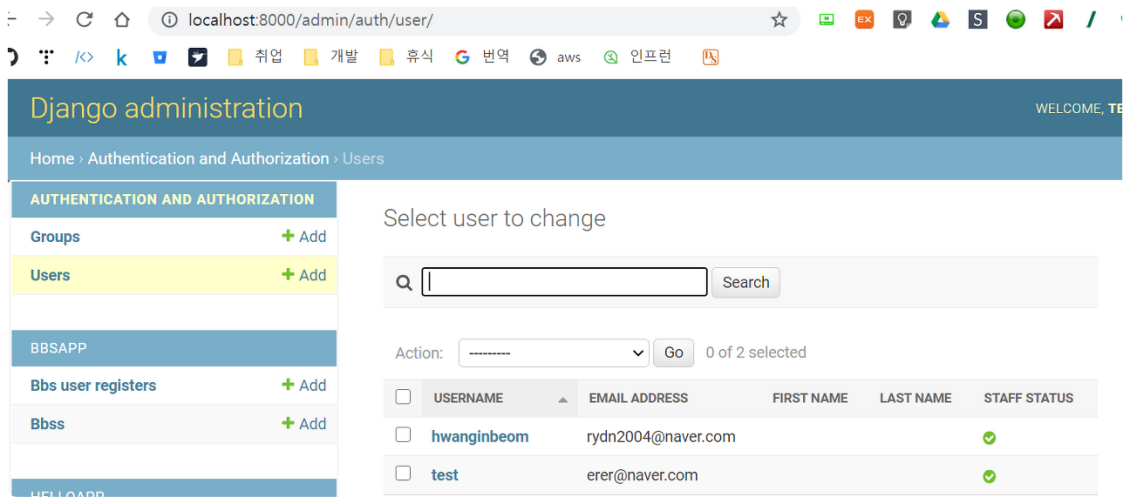
```
1 from django.shortcuts import render, redirect
2
3 # Create your views here.
4 from .models import BbsUserRegister
5
6
7
8 def loginForm(request):
9     return render(request, 'login.html')
10
11 def registerForm(request):
12     return render(request, 'join.html')
13
14 def register(request):
15     if request.method == 'POST':
16         id = request.POST['id'] # 값을 받아오는 작업
17         pwd = request.POST['pwd']
18         name = request.POST['name']
19         register = BbsUserRegister(user_id=id, user_pwd=pwd, user_name=name) # 해당 클래스의 형태로 객체를 만드는 작업
20         register.save() # 객체를 models에 저장한다.
21         return redirect('loginForm') # registerForm url에 이름으로 지정해놓은 loginForm으로 이동하게 된다.
```

이렇게 view 를 구성하고



계정을 생성 할 경우 기존 login으로 이동하게 된다.





admin으로 가서 user 를 확인 할 경우 아이디가 생성 된 걸 볼수있다.

## 'Django' 카테고리의 다른 글

[Django] #8 - Django 세션으로 데이터 넘기기 / 로그인 구현 / 로그인시 로그...

[Django] #7 - header / footer 를 통한 templates 구현

**[Django] #6 - Django 를 통한 사용자 등록 구현하기**

[Django] #5 - Django 를 통한 static 사용하기

[Django] #4 - Django ORM을 통한 데이터 관리 예제

[Django] #3 - Django ORM을 통한 데이터 관리

django 사용자 등록

장고 사용자 등록

장고 아이디 만들기



꾸까꾸

혼자 끄적끄적하는 블로그 입니다.

