

Django Authentication System



- ❖ 학습해야 할 내용
 - ✓ Django Accounts

1. User Model BooleanField

django에서 기본적으로 사용하는 User 모델은 AbstractUser 모델을 상속받아 정의된다.

```
class User(AbstractUser):
    """
    Users within the Django authentication system are represented by this
    model.
    Username and password are required. Other fields are optional.
    """
    class Meta(AbstractUser.Meta):
        swappable = 'AUTH_USER_MODEL'
```

- 아래의 models.py를 참고하여 User 모델에서 사용할 수 있는 컬럼 중 BooleanField로 정의된 컬럼을 모두 작성하시오.

<https://github.com/django/django/blob/master/django/contrib/auth/models.py>

2. username max length

django에서 기본적으로 사용하는 User 모델의 username 컬럼이 저장할 수 있는 최대 길이를 작성하시오.

3. login validation

단순히 사용자가 '로그인 된 사용자인지'만을 확인하기 위하여 User 모델 내부에 정의된 속성의 이름을 작성하시오.



Django Authentication System

4. Login 기능 구현

다음은 로그인 기능을 구현한 코드이다. 빈 칸에 들어갈 코드를 작성하시오.

```
from django.contrib.auth.forms import __ (a) __
from django.contrib.auth import __ (b) __ as auth_login

def login(request):
    if request.method == 'POST':
        form = __ (a) __ (request, request.POST)
        if form.is_valid():
            auth_login(request, __ (c) __)
            return redirect('accounts:index')
    else:
        form = __ (a) __ ()
        context = {
            'form': form,
        }
        return render(request, 'accounts/login.html', context)
```

5. who are you?

로그인을 하지 않았을 경우 template에서 user 변수를 출력했을 때 나오는 클래스의 이름을 작성하시오.

6. 암호화 알고리즘

Django에서 기본적으로 User 객체의 password 저장에 사용하는 알고리즘, 그리고 함께 사용된 해시 함수를 작성하시오.

7. Logout 기능 구현

로그아웃 기능을 구현하기 위하여 다음과 같이 코드를 작성하였다. 로그아웃 기능을 실행 시 문제가 발생한다고 할 때 그 이유와 해결 방법을 작성하시오

```
def logout(request):
    logout(request)
    return redirect('accounts:login')
```