

01. 프로젝트명

Django를 사용한 장고스타그램 웹 플랫폼 만들기

02. 프로젝트 설명 (200자)

웹 개발을 위한 Python 프레임워크인 Django 사용법을 익힙니다.
 웹 게시판 기능을 함께 만들어가며 Django의 기본 활용법을 배웁니다.
 Django는 웹 사이트를 백엔드부터 프론트엔드까지 매끄럽게 만들 수 있는 웹 프레임워크입니다.
 웹 사이트의 기본인 회원 관리부터 게시물 보기 및 작성, 데이터베이스 생성과 어드민 페이지 제작을 담은 '장고스타그램' 웹 플랫폼을 Django를 통해 제작합니다.

03. 프로젝트 과제 상세 : 수강생들이 프로젝트를 진행할 때, 유념해야 할 것들입니다.

기준	과제 상세	
프로젝트 구성 단계	1	djangostagram 장고 프로젝트를 생성하세요
사용자 관리	2	아이디, 이메일, 비밀번호, 가입일을 저장하는 Dsuser 모델을 만드세요. 가입일에는 자동으로 현재 DateTime이 저장되어야 합니다
	3	비밀번호는 hashers를 사용해 암호화 해서 저장해주세요
	4	base.html 템플릿을 만들어 모든 페이지에서 상속받아 보여주세요
공통 템플릿	5	Bootstrap을 cdn으로 넣은 후 bootswatch 홈페이지에서 마음에 드는 테마를 다운로드하여 static 경로로 넣어주세요
	6	(상단 네비게이션) 로그인인 안된 상태라면 '타임라인', '회원가입', '로그인' 메뉴를 보여주세요
	7	(상단 네비게이션) 로그인인 된 상태라면 '타임라인', '로그아웃', '글 작성' 메뉴를 보여주세요
	8	각 메뉴를 누르면 해당 페이지로 이동합니다
회원가입 페이지	9	'user/register' url에 들어가면 회원가입 페이지를 보여주세요
	10	아이디, 이메일, 비밀번호, 비밀번호 확인을 받는 회원가입 페이지를 만드세요
	11	하나라도 입력을 하지 않고 폼을 전송한다면 '모든 값을 입력해야 합니다' 라는 여러 메시지를 보여주세요
	12	비밀번호와 비밀번호 확인 값이 다르면 '비밀번호가 다릅니다' 라는 여러 메시지를 보여주세요
	13	회원 생성이 완료되면 메인 화면으로 이동시켜주세요
로그인 페이지	14	'user/login' url에 들어가면 로그인 페이지를 보여주세요
	15	forms.py에 LoginForm을 만들고 이를 활용해 아이디, 비밀번호를 받는 로그인 페이지를 만드세요
	16	반복문을 이용해 template에서 form field를 가져와서 그려주세요
	17	잘못된 아이디로 로그인을 시도하면 '아이디가 없습니다' 여러 메시지를 보여주세요
	18	잘못된 비밀번호로 로그인을 시도하면 '비밀번호를 틀렸습니다' 여러 메시지를 보여주세요
	19	로그인이 성공하면 메인 화면으로 이동시켜주세요
포스트 작성 페이지	20	작성자, 이미지 주소, 내용, 작성일, 태그를 저장하는 Post 모델을 만드세요. 작성자는 Dsuser 모델과 연결하고 태그는 ag 모델과 다대다 연결을 해주세요
	21	'upload' url에 들어가면 포스트 작성 페이지를 보여주세요
	22	로그인이 되어있지 않다면 로그인 페이지로 리다이렉트 해주세요
	23	form.py에 PostForm을 만들고 이를 활용해 제목, 내용, 태그를 받는 포스트 작성 페이지를 만드세요
	24	이미지 주소와 내용은 필수값으로 하고 입력하지 않으면 에러메시지를 보여주세요
	25	내용 필드는 여러 줄을 입력할 수 있는 textarea로 보여주세요 (form widget 활용)
	26	태그는 콤마(,)를 통해 여러 개 저장할 수 있도록 해주세요 (e.g. '강아지, 고양이')
타임라인 페이지 (메인 화면)	27	'/' url에 들어가면 타임라인 페이지를 보여주세요
	28	포스트를 생성일 역순으로 정렬해서 카드형식으로 보여주세요 (부트스트랩의 Card 활용)
	29	각 포스트에 사진, 작성자, 작성일을 보여주세요
	30	한 페이지에 포스트가 4개씩 보이도록 하고 페이지네이터를 넣어주세요
	31	로그인이 되어있는 상태라면 상단에 'Hello, {아이디}' 텍스트를 보여주세요
포스트 상세 페이지	32	타임라인 페이지에서 포스트 카드를 누르면 해당 포스트 상세페이지로 이동합니다
	33	'/포스트/{포스트id}' url에 들어가면 해당 포스트 상세 페이지를 보여주세요
	34	사진, 작성자, 작성일, 태그를 보여주세요
	35	타임라인 페이지로 돌아가는 버튼을 만들어주세요
	36	잘못된 포스트id로 들어가면 404 페이지를 보여주세요
기타 앱 설정	37	사용자, 포스트, 태그 앱을 Admin에서 볼 수 있도록 등록하세요
	38	사용자 모델은 리스트에서 아이디와 비밀번호를 바로 볼 수 있도록 list_display를 변경하세요
프로젝트 배포 단계	39	setting.py에 pythonanywhere를 위한 ALLOWED_HOSTS와 STATIC_ROOT 설정을 하세요

04. 점검 기준표 : 아래 체크리스트 기준을 참고하여 리뷰를 진행해주세요.

기준	점수 기준	점검 상세
프로젝트 구성 단계	<input type="checkbox"/>	djangostagram 장고 프로젝트를 생성했다
사용자 관리	<input type="checkbox"/>	model.py에 Dsuser 모델을 만들고 아이디, 이메일, 비밀번호, 가입일 필드를 생성했다. 가입일에는 auto_now_add=True 옵션을 넣었다
	<input type="checkbox"/>	비밀번호를 저장할 때 from django.contrib.auth.hashers import make_password 를 통해 평문이 아닌 암호화하여 저장했다
공통 템플릿	<input type="checkbox"/>	template/base.html 을 만들고 모든 템플릿에서 {% extends 'base.html' %} 로 상속받아서 사용한다
	<input type="checkbox"/>	base.html에 bootstrap css, js cdn 파일을 넣었다
	<input type="checkbox"/>	static 폴더를 생성해 bootswatch 테마 css 파일을 다운로드해 넣고 base.html 에 static 경로를 사용해 집어넣었다
	<input type="checkbox"/>	(상단 네비게이션) {% if request.session.user %} 를 활용해 로그인인 안된 상태에 '타임라인', '회원가입', '로그인' 메뉴를 보여준다
	<input type="checkbox"/>	(상단 네비게이션) {% if request.session.user %} 를 활용해 로그인인 된 상태라면 '타임라인', '로그아웃', '글 작성' 메뉴를 보여준다
회원가입 페이지	<input type="checkbox"/>	각 메뉴에 href="{% url 'url이름' %}"을 넣어 메뉴를 누르면 해당 페이지로 이동한다
	<input type="checkbox"/>	'user/register' url과 회원가입 템플릿을 연동했다
	<input type="checkbox"/>	아이디, 이메일, 비밀번호, 비밀번호 확인을 빠짐없이 받는 회원가입 템플릿을 만들었다
	<input type="checkbox"/>	아이디, 이메일, 비밀번호, 비밀번호 확인 중 한 필드라도 모자라면 '모든 값을 입력해야 합니다' 에러 메시지를 보여준다
	<input type="checkbox"/>	비밀번호와 비밀번호 확인 값이 다르면 '비밀번호가 다릅니다' 라는 에러 메시지를 보여준다
로그인 페이지	<input type="checkbox"/>	값을 저장해 Dsuser 모델을 생성하고, 성공시 메인 화면으로 redirect 한다
	<input type="checkbox"/>	'user/login' url과 로그인 템플릿을 연동했다
	<input type="checkbox"/>	form.py에 LoginForm을 만들고 이를 활용해 아이디, 비밀번호를 받는 로그인 페이지를 만들었다
	<input type="checkbox"/>	{% for field in form %} 를 이용해 template에서 form field를 가져와서 그려주었다
	<input type="checkbox"/>	Dsuser.DoesNotExist 에러가 나면 '아이디가 없습니다' 에러 메시지를 보여준다
포스트 작성 페이지	<input type="checkbox"/>	form django.contrib.auth.hashers import check_password 를 통해 암호화 된 비밀번호를 확인하고 잘못되었으면 에러 메시지를 보여준다
	<input type="checkbox"/>	로그인이 성공하면 메인 화면으로 리다이렉트 시킨다
	<input type="checkbox"/>	작성자, 이미지 주소, 내용, 작성일, 태그를 저장하는 Post 모델을 만들었다. 작성자는 ForeignKey로 Dsuser와 연결, 태그는 ManyToManyField로 Tag모델과 연결했다
	<input type="checkbox"/>	'upload' url과 포스트 작성 템플릿을 연동했다
	<input type="checkbox"/>	로그인이 되어있지 않다면 로그인 페이지로 리다이렉트 한다
타임라인 페이지 (메인 화면)	<input type="checkbox"/>	form.py에 PostForm을 만들고 이를 활용해 제목, 내용, 태그를 받는 포스트 작성 페이지를 만들었다
	<input type="checkbox"/>	이미지 주소와 내용에 error_messages={'required': '에러메시지'}를 활용해 필수값 설정 후 에러 메시지를 보여주었다
	<input type="checkbox"/>	내용 필드를 widget=forms.Textarea를 사용해 textarea로 보여주었다
	<input type="checkbox"/>	태그를 콤마(,)로 split하여 각각 Tag 오브젝트에 저장하였다
	<input type="checkbox"/>	'/' url과 타임라인 템플릿을 연동했다
포스트 상세 페이지	<input type="checkbox"/>	포스트를 order_by('-생성일')로 정렬하고 부트스트랩의 Card를 활용해 카드 형식으로 보여주었다
	<input type="checkbox"/>	각 포스트에 사진, 작성자, 작성일을 보여주었다
	<input type="checkbox"/>	Paginator를 4로 설정해 한 페이지에 포스트가 4개씩 보이도록 하고 템플릿에 페이지네이션을 설정하였다
	<input type="checkbox"/>	로그인이 되어있는 상태라면 상단에 'Hello, {아이디}' 텍스트가 보인다
	<input type="checkbox"/>	포스트 카드에 <a href="{% url 'post' pk=post.pk %}" 링크를 넣어 포스트 상세 페이지로 갈 수 있도록 했다
기타 앱 설정	<input type="checkbox"/>	'/포스트/{포스트id}' url과 포스트 상세 템플릿을 연동했다
	<input type="checkbox"/>	사진, 작성자, 작성일, 태그를 보여준다
	<input type="checkbox"/>	타임라인 페이지로 돌아가는 버튼을 만들었다
	<input type="checkbox"/>	get_object_or_404 를 활용해 잘못된 포스트 id로 들어가면 404 에러를 보여준다
	<input type="checkbox"/>	각 앱의 admin.py 에 admin.site.register(모델명)으로 admin에 모델을 등록했다
프로젝트 배포 단계	<input type="checkbox"/>	dsuser/admin.py 에 list_display = ('아이디', '비밀번호') 를 넣었다
	<input type="checkbox"/>	settings.py의 ALLOWED_HOSTS에 '아이디.pythonanywhere.com'를 넣고 STATIC_ROOT에 os.path.join(BASE_DIR, 'static')를 넣었다
	<input type="checkbox"/>	배포된 url이 정상 작동한다