**Learned**

* Writing your first Django app, part 4 (<https://github.com/naya-h2/Django-Study/tree/week4>)

**Error-log**

**⚠️ Choice 데이터 DB에 저장**

* vote view를 구현하는데, question 데이터만 있고 연결된 choice 데이터가 없어서 당황스러웠다.
* 다시 python shell로 q(Question-1)에 대해서 choice\_set을 만들었는데, 얘를 db에 저장하려면 q.choice\_set.save()를 하면 될 줄 알았는데 안됐다!
* 알고 보니까 q에 연결된 choice인 거라서 **q 자체를 save**하면 해야 하는 거였다,, 어쩌다 보니 같은 choice set을 두 번이나 저장해버렸다,,

**etc**

* 장고의 페이지 레이아웃이 로컬 폴더에서 직관적으로 확인할 수 있는게 아니라, [urls.py](http://urls.py) 파일에 들어가서 코드로 직접 확인을 해야 한다는 게 어렵고 헷갈린다…
* 내가 사용하던 html은 페이지 구조만 설정하고, 로직은 js script를 따로 넣는 방식이었는데 장고에서 사용하는 html에는 직접적으로 로직이 들어가 있어서 코드를 해석하기가 훨씬 헷갈리는 것 같다.
* 제네릭 뷰를 처음 공식 레퍼런스에서 보고 이게 무슨 소리지..?! 내가 직접 컴포넌트처럼 공통 부분을 따로 만들어서 재활용하는건가? 생각했는데 다른 블로그를 찾아보니까 장고에서 제공해주는 템플릿같은거였다!
  + 그 템플릿을 살짝만 바꾸면 되니까 코드 양을 줄이기 위해서 사용한다고 한거였구나 ?!

**Think**

* 특이하게 post 메소드로 설정하는 부분이 로직이랑 관련되어 있다고 생각해서 view의 함수 내부에 있을거라고 생각했는데 html 파일에서 설정해주는 게 신기했다!
* html에서 어떤 메소드를 사용할 지 결정하고, view에서는 그 응답을 받아서 처리만 하는 방식이 온전히 백엔드스럽지는 않은 개발인 것 같다고 느껴졌다.
  + 이런 의미에서 장고가 풀스택 개발 지원 언어인걸까?!
* 솔직히 장고의 구체적인 동작 방식을 잘 모르겠다!