

# week8 실습

## ▼ 오류해결

1. 만약 장고가 없다는 오류가 뜬다면



```
pip install django
```

2. 만약 `ModuleNotFoundError: No module named 'corsheaders'` 라는 오류가 뜬다면 터미널에서



```
pip install django-cors-headers
```

명령어 실행

이후 `frontendSeminar8/settings.py`의 `INSTALLED_APPS` 에 'corsheaders' 추가 (없다면)

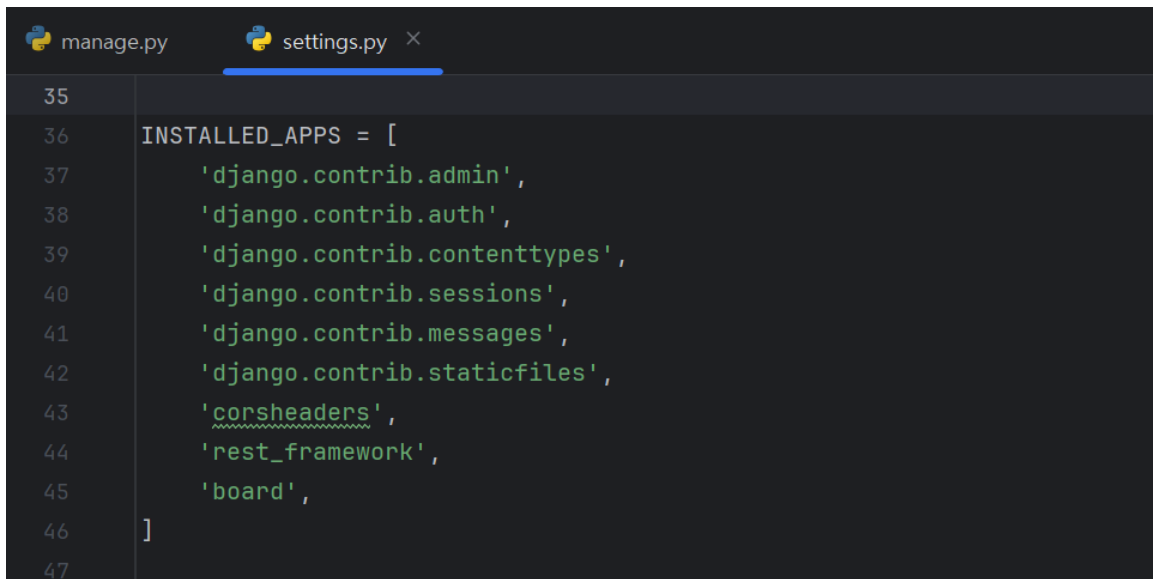
3. 만약 `ModuleNotFoundError: No module named 'rest_framework'` 라는 오류가 뜬다면 터미널에서



```
pip install djangorestframework
```

명령어 실행

이후 마찬가지로 `frontendSeminar8/settings.py`의 `INSTALLED_APPS` 에 'rest\_framework' 추가 (없다면)



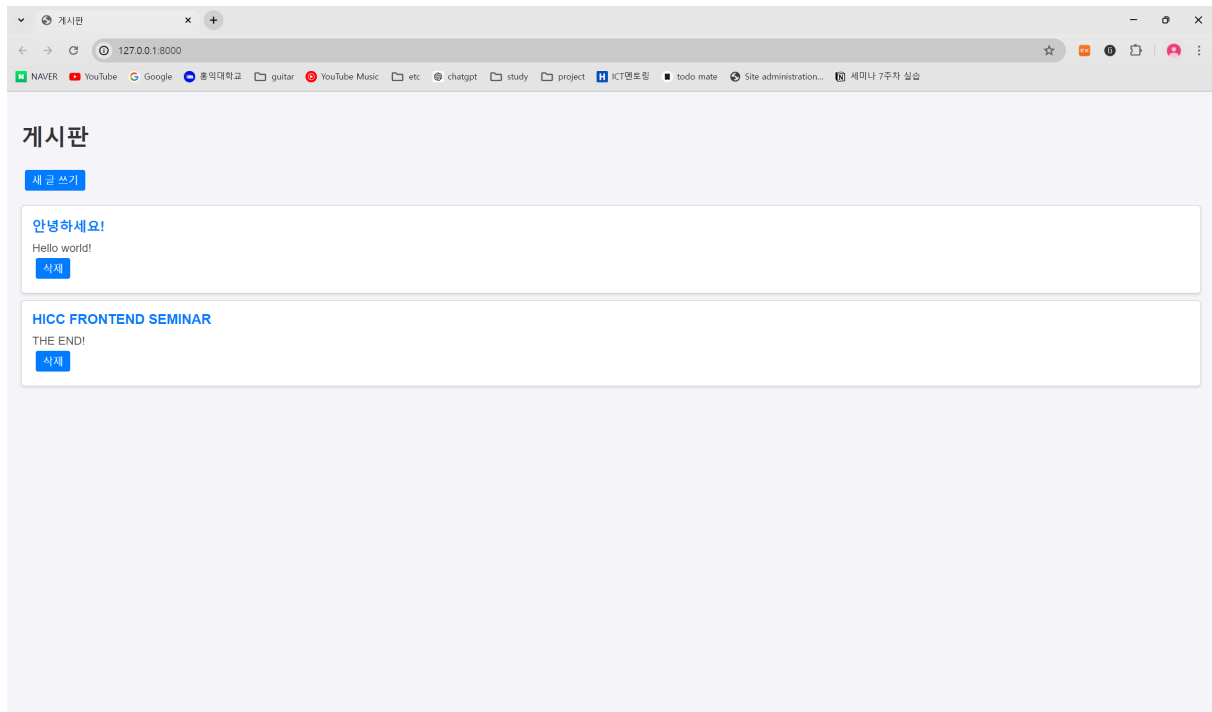
```
35
36 INSTALLED_APPS = [
37     'django.contrib.admin',
38     'django.contrib.auth',
39     'django.contrib.contenttypes',
40     'django.contrib.sessions',
41     'django.contrib.messages',
42     'django.contrib.staticfiles',
43     'corsheaders',
44     'rest_framework',
45     'board',
46 ]
47
```

### ▼ 서버 실행 방법

.zip파일 압축을 풀고 해당 파일을 터미널로 들어가서 경로를 manage.py 가 있는 곳까지 옮겨놓기

이후 `python manage.py runserver` 로 서버 실행

### • 결과물



- 실습 (파일 별로 `<!-- to write -->`, `/* to write*/`, `//to write` 에 해당하는 부분을 채우기)

## 1. index.html

- html의 head 태그에서 필수로 들어가야하는 태그를 작성
- '새 글 쓰기'에 해당하는 부분의 버튼 만들기 ( id는 "new-post-button" )
- 다음 그림처럼 제목은 한 줄만 입력받고, 내용은 여러 줄을 입력받을 수 있게 되어 있다. 주어진 속성을 활용하여 아래와 같이 만들어보기

## 2. styles.css

- 다음 조건들을 보고 해당하는 코드를 작성해보기
  - 해당 웹의 폰트는 Arial, 만약 Arial 이 없을 경우 sans-serif를 지원한다.
  - 해당 웹의 배경 색은 #f4f4f9 이다.
  - 해당 웹의 기본 마진은 0이다.
  - 해당 웹의 기본 padding은 20px이다.
- post-form 이라는 id를 갖는 div태그의 스타일들은 다음과 같을 때 해당하는 코드를 작성해보기 (코드에 선언문은 제공되어 있음)

```
margin-top: 20px;
padding: 20px;
background: #fff;
border: 1px solid #ccc;
border-radius: 5px;
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
```

## 3. scripts.js

- 게시물을 저장하거나 쓰다가 취소하기 위해서는 '새 글 작성', '저장', '취소' 버튼을 눌러야 할 때, 해당하는 버튼의 id를 찾아서 event 발생시켜보기  
각자 실행할 함수의 이름은 'showForm', 'addPost', 'hideForm'이다.

- b. `hideForm()` 함수는 해당 블록을 안보이게 해주는 함수이다. 위의 `showForm()`을 참고하고, 검색을 통해 특정 요소를 숨기는 기능을 만들어보기
- c. 게시물을 서버에 보내는 함수를 작성해보기
  - i. `title`과 `content`의 `id`를 받아서 둘 다 존재해야 게시물을 서버에 보낼 수 있음
  - ii. 보낼 `post` 구조체는 `{title, content}`로 이루어져 있음
  - iii. `fetch API`를 사용해서 데이터를 전송하기
  - iv. `url`은 ' `http://localhost:8000/api/posts/` '
  - v. 요청 본문(`body`)에 포함할 데이터를 `JSON`형식으로 변환, 자바스크립트 객체를 변환할 때는 `JSON.stringify` 함수 사용
  - vi. 요청을 보내고 `id`가 'post-form'인 글 작성 폼을 `.reset()`를 통해서 `reset` 시켜주기
  - vii. 화면에 떠있는 글 작성 폼을 안보이게 하고, 게시물을 불러옴
  - viii. `error`가 있다면 `console.error()`를 사용하여 에러 메시지 띄워주기