Python Django

4강

원래는 video를 예시로 계속 설명했지만..
이제는 한계가 왔습니다 여러분 ㅠㅠㅠ
우리 같이 음악 스트리밍 프로그램을 만들어봐요~

Python Django

1. 세부 템플릿 만들기

detail.html을 수정!

```
<img src="{{ album.album_logo }}">
<h1>{{ album.album_title }}</h1>
<h3>{{ album.artist }}</h3>

<ul>
    {% for song in album.song_set.all %}
        <li>{{ song.song_title }} - {{ song.file_type }}</li>
    {% endfor %}
</ul>
```

Python Django

2. Url 유동적으로 만들기

현재 index.html 의 href 부분을 보면 music 이 바로 url에 명시되어 있어 이 앱에서 밖에 적용을 못함

```
{% for album in all_albums %}
    <li><a href="/music/{{ album.id }}/">{{ album.album_title }}</a></li>
{% endfor %}
```

따라서 이 url을 어디든 사용할 수 있도록 유동적으로 바꾸는 코딩을 할 것임!

이것을 위해 우리가 해줬던 것이 바로 url에 naming을 해준 것 !

```
urlpatterns = [
    # /music/
    url(r'^$', views.index, name = 'index'),

    #/music/IDNum/
    url(r'^(?P<album_id>[0-9]+)/$', views.detail, name = 'detail'),

]
```

각 url 형태에 그 이름이 대응된다고 생각하면 쉽다
/music/ -> index
/music/albumid/ -> detail

Python Django

2. Url 유동적으로 만들기

이걸 이용해서 index.html의 hardcoded url을 지워 줄 것임

```
{% for album in all_albums %}
    <li><a href="{% url 'detail' album.id %}">{{ album.album_title }}</a></li>
{% endfor %}
```

* 각 앱이 해당 url의 이름을 가지고 있는 경우
(ex/ music앱도 video앱도 모두 detail이란 이름을 가진 url이 정의되어 있는 경우)
-> 모든 앱에서 하나의 앱에서 정의된 url의 형식을 가져오게 됨!

이것을 막기 위해 namespace 를 이용한다
해당 앱의 url.py에 이렇게 작성 -> app_name = 'music'
후에 html파일에 이렇게 수정

```
{% for album in all_albums %}
    <li><a href="{% url  'music:detail' album.id %}">{{ album.album_title }}</a></li>
{% endfor %}
```

Python Django

3. Http404 error shortcut

Detail function 안의 404에러를 더 간단히 써줄 수 있는 방법이 있다

1) Shortcuts 에서 get_object_or_404 를 import

```
from django.shortcuts import render, get_object_or_404
```

2) Detail 함수 부분을 다음과 같이 바꿔준다

```
def detail(request,album_id):
    album = get_object_or_404(Album, pk=album_id)
    return render(request,'music/detail.html',{'album': album})
```

-> 이 자체가 존재하면 album을 넘겨주고, 아니면 404에러를 띄울 수 있는 메서드로 작용

Python Django

4. 새로운 View Form 만들기

노래에 대해 favorite을 추가 할 수 있는 기능을 웹에 넣어보자!

-〉 라디오버튼으로 체크, favorite 버튼을 누르면 곡명 옆에 별이 뜨게!

1) Song의 model 요소에 is_favorite 추가

```
class Song(models.Model):
    album = models.ForeignKey(Album, on_delete=models.CASCADE)
    file_type = models.CharField(max_length=10)
    song_title =models.CharField(max_length=200)
    is_favorite = models.BooleanField(default=False)
```

-〉 곡명의 favorite 설정 유무를 Boolean 으로 나타냄

후에 terminal에서 makemigration, migrate

Python Django

4. 새로운 View Form 만들기

2) Favorite url 만들기

Favorite 함수의 url은 favorite 버튼을 눌렀을 경우 다시 그 사이트 페이지를 띄우도록 만들어져야 함
-> 따라서 이에 대한 특별한 view가 존재하지 않고 그냥 그 기능만 실행 하도록 만들어 주면 됨 !

일단 url.py 에 favorite 기능을 실행할 url만듬

```
urlpatterns = [
    # /music/
    url(r'^$', views.index, name = 'index'),

    #/music/IDNum/
    url(r'^(?P<album_id>[0-9]+)/$', views.detail, name = 'detail'),

    #/music/IDNum/favorite/
    url(r'^(?P<album_id>[0-9]+)/favorite/$', views.favorite, name = 'detail'),

]
```

Python Django

4. 새로운 View Form 만들기

3) Favotie 동작 가능하도록 detail.html 수정

```
<h3>{{ album.artist }}</h3>

{% if error_message %}
    <p><strong>{{ error_message }}</strong></p>
{% endif %}

<form action = "{% url 'music:favorite' album.id %}" method="post">
    {%  csrf_token %}
    {%  for song in album.song_set.all %}
        <input type="radio" id="song{{ forloop.counter }}" name="song" value="{{ song.id }}" />
        <label for="song{{ forloop.counter }}" >
            {{ song.song_title }}
            {% if song.is_favorite %}
                <img src="http://i.imgur.com/b9b13Rd.png"/>
            {% endif %}
        </label><br>
    {% endfor %}
     <input type="submit" value="Favorite">
</form>
```

//선택하지 않았을 경우 경고 문구

//favorite를 한 노래 옆에 별 생성
Csrf token -> 보안을 위해 존재하는 것

Python Django

4. 새로운 View Form 만들기

4) view.py 에서 favorite 함수 선언

```
def favorite(request, album_id):
    album = get_object_or_404(Album, pk=album_id)
    try:
        selected_song = album.song_set.get(pk=request.POST['song'])
    except (KeyError, Song.DoesNotExist):
        return  render(request,'music/detail.html',{
            'album': album,
            'error_message' : "You did not select a valid song",
        })
    else:
        selected_song.is_favorite =True
        selected_song.save()
        return render(request, 'music/detail.html', {'album': album})
```

Song 의 value를 가져옴

// 노래를 선택하지 않았을 경우
오류 뜨도록 만듬

//오류가 없을 경우에는 is_favorite 항목이
true로 바뀌고 다시 페이지 표시

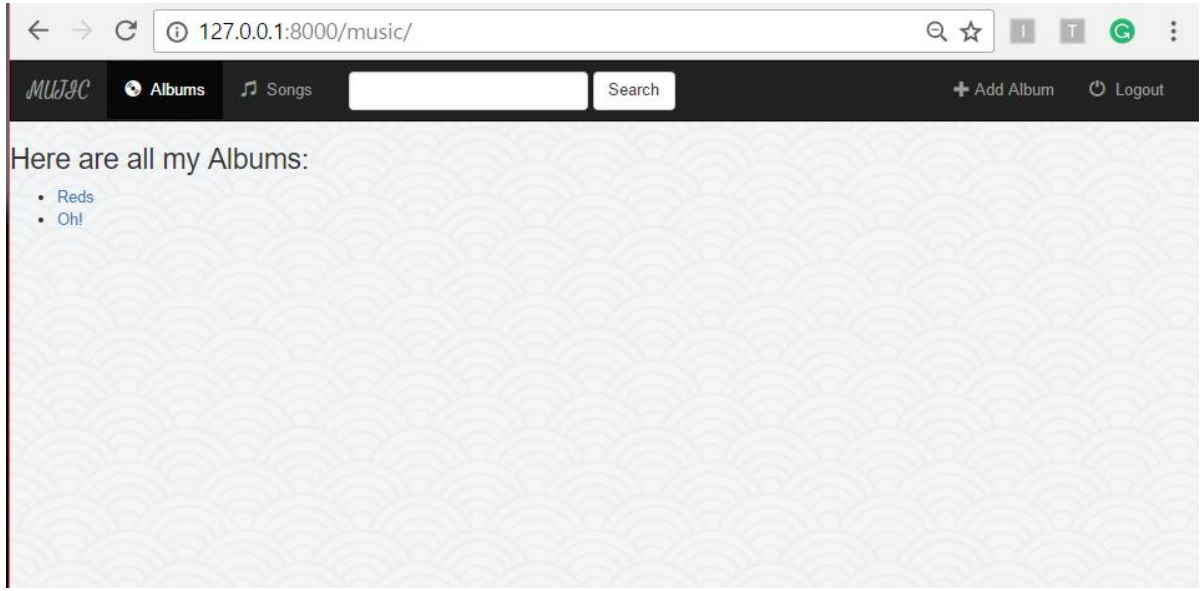
Python Django

5. Bootstrap과 Static File

이제부터는 부트스트랩을 사용하여 배경설정 / 네비게이터 바 / 폰트 설정을 하겠습니다

Bootstrap : 부트스트랩은 동적인 웹 사이트 및 웹 응용 개발을 위한 프론트엔드 프레임워크로, 입력 창, 버튼, 네비게이션 및 기타 구성물, 각종 레이아웃 등을 HTML 및 CSS 기반의 디자인 템플릿으로 제공하며 추가적인 자바스크립트 확장들도 포함한다. **[네이버 지식백과]** 부트스트랩 [Bootstrap] (두산백과)

Static file : 그 웹의 디자인을 위해 필요한 고정된 파일 -〉 아래의 배경 사진! 같은 것



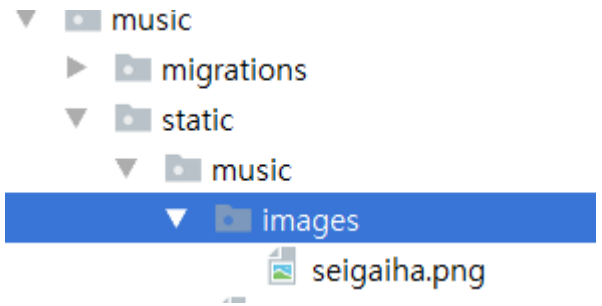
Python Django

https://www.toptal.com/designers/subtlepatterns/
-> 가지고 오고 싶은 웹 배경을 가져오세요

5. Bootstrap과 Static File

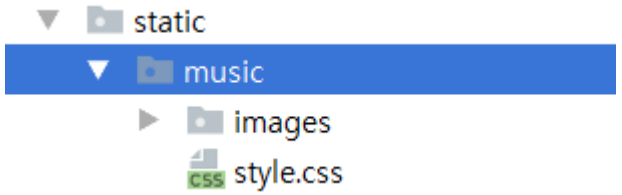
1)배경 띄우기

아까의 static file에 대한 설명 처럼 그 파일을 담고 있어야 할 폴더를 만들어 줘야함



Music app파일에 이렇게 static/music/images 디렉토리를 생성해주고
가지고 오고 싶은 웹 배경을 images폴더 안에 넣어주기!

그리고 배경 적용을 위해 static/music 폴더 안에 css 파일을 만들어 주고, css 작성




```
body{
    background: white url("images/seigaiha.png");
}
```

Python Django

5. Bootstrap과 Static File

1) 배경 띄우기

후에 css파일을 index.html에 연결 -> 앞머리에 아래 코드 추가

```
<!--Loads the path to your static files-->
{%  load staticfiles %}
<link rel="stylesheet" type="text/css" href="{%  static  'music/style.css' %}" />
```

2) Navigation bar 만들기

저 코드 위에 추가로 부트스트랩 링크를 연결시켜줌

```
<!--Loads the path to your static files-->
{%  load staticfiles %}
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="{%  static  'music/style.css' %}" />
```

링크를 연결시켜주면 부트스트랩 툴을 사용 할 수 있음!

Python Django

5. Bootstrap과 Static File

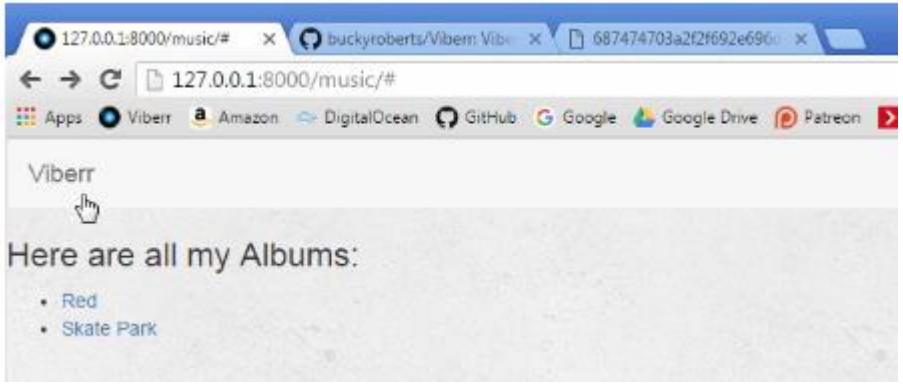
2) Navigation bar 만들기

Index.html 바로 아래에 추가

```
<nav class="navbar navbar-default">
    <div class="container-fluid">

        <!-- Logo -->
        <div class="navbar-header">
            <a class="navbar-brand" href="{% url 'music:index' %}">Viberr</a>
        </div>

    </div>
</nav>
```



// 해당 navigation bar 안에 이름 -〉 링크
〈a〉 -〉 다른 웹페이지를 연결할 때 쓰는 태그

Python Django

https://fonts.google.com/

Fonts.Googleapies -〉 폰트관련 외부파일
나머지 두개는 자바스크립트, jquery
-〉 웹사이트를 줄이거나 늘릴 때 더 예쁘게 만들어 줄 수 있음
-〉 다양한 아이콘 제공

5. Bootstrap과 Static File

2) Navigation bar 만들기 - menu

다채로운 홈페이지를 위해 새로운 부트스트랩 + 자바스크립트 추가

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<link href="https://fonts.googleapis.com/css?family=Satisfy" rel="stylesheet">
<link rel="stylesheet" type="text/css" href="{%  static  'music/style.css' %}" />
<script src = "http://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
```

후에 navbar-default -〉 inverse (그냥 색을 흰색-〉검정색 변경) 후 아래 코드 작성

```
<nav class="navbar navbar-inverse">
    <div class="container-fluid">

        <!-- Header -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle = "collapse" data-target="#topNavBar">
                <span class ="icon-bar" ></span>
                <span class ="icon-bar" ></span>
                <span class ="icon-bar" ></span>
            </button>
            <a class="navbar-brand" href="{%  url 'music:index' %}">MUJIC</a>
        </div>
-
```

// 메뉴 버튼 생성

Python Django

5. Bootstrap과 Static File

2) Navigation bar 만들기 - items

이제 네비게이션 바에 기능들을 추가 할 것
아래 새로운 division 생성하여 album, song 열람 할 수 있는 페이지 설정 만듬

```
<!-- Item -->
<div class="collapse navbar-collapse" id ="topNavBar">
    <ul class="nav navbar-nav">
        <li class="active">
            <a herf = "{%  url 'music:index' %}">
                <span class="glyphicon glyphicon-cd" aria-hidden="true"></span>  Albums
            </a>
        </li>
        <li class="">
            <a herf =  "#">
                <span class="glyphicon glyphicon-music" aria-hidden="true"></span> Songs
            </a>
        </li>
    </ul>
</div>
```

// 는 아이콘과 글자 간의 약간의 간격을 주도록 함

// class='active' -> 현재 선택된 창이라는 것을 알려줘서 글자가 더 하얗고 바탕은 까맣게!
//저 id는 모든 아이콘을 표시 할 수 없을 경우 메뉴 창이 toggle형식으로 아래로 떨어지는 메뉴바로 수용하게 되는데 그 들어갈 요소들을 id로서 지정해준다
-> <button 안에 있는 data-target='#topNavBar' 과 대응된다고 생각하면 됨

Python Django

5. Bootstrap과 Static File

2) Navigation bar 만들기 - search, add album, logout

이어서 아래에 새로운 〈form〉으로 search 기능 추가

```
<form class="navbar-form navbar-left" role="presentation" method="get" action="#">
    <div class = "form-group">
        <input type="text" class="form-control" name="q" value="">
    </div>
    <button type="submit" class="btn btn-default">Search</button>
</form>
```

// 위에서 했던 기능들과 같은 div에 넣어 줘야함

후에 나머지 기능 ul div로 만들어줌 - 노래추가, 로그아웃

```
<ul class="nav navbar-nav navbar-right">
    <li class="">
        <a herf = "#">
            <span class="glyphicon glyphicon-plus" aria-hidden="true"></span> Add Album
        </a>
    </li>
     <li class="">
        <a herf = "#">
            <span class="glyphicon glyphicon-off" aria-hidden="true"></span>  Logout
        </a>
    </li>
</ul>
```

Python Django

5. Bootstrap과 Static File

2) Navigation bar 만들기 - logo 글씨 바꾸기 / 모양 바꾸기

Style.css 파일에서 내용 추가

```
.navbar{
    border-radius: 0;
}

.navbar-brand{
    font-family: 'Satisfy', cursive;
}
```

그러면 완성!!!!!!!!까아아ㅏ