2014-2 EEE3545-01 Operating System

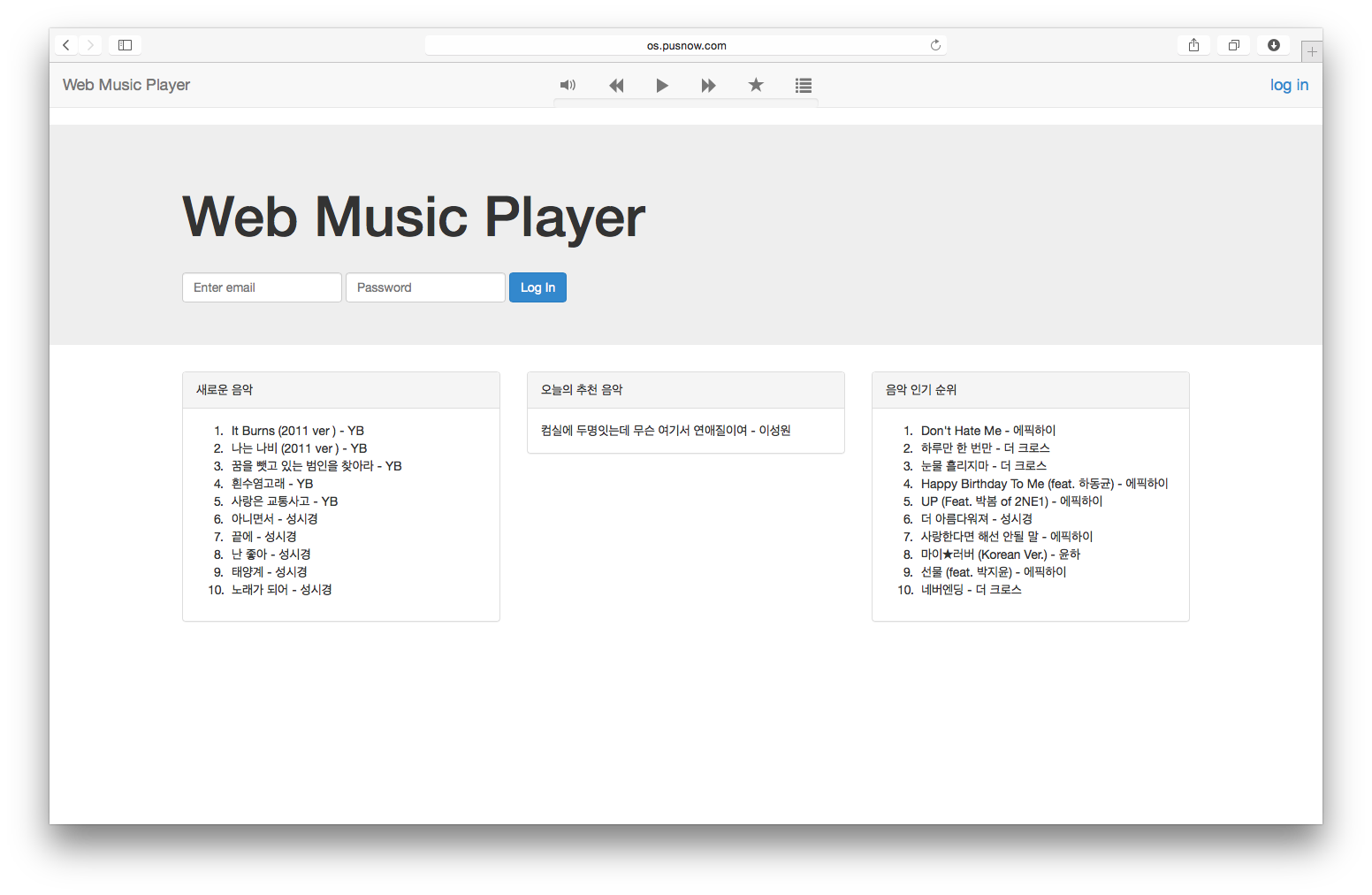
Web Streaming Player

**11조 윤원섭, 이성원, 양형규, 이영조**

**담당 조교 : 전동현, 김정환, 강지우**

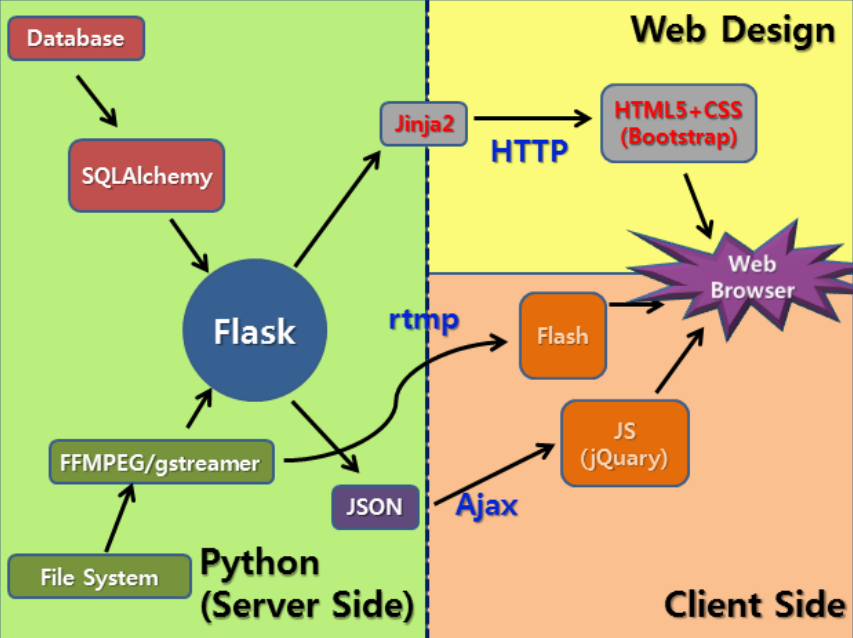
# 프로젝트 설명 및 개요

이번 프로젝트는 웹에서 음악을 스트리밍 방식으로 재생하는 Player을 구현하는 것이었다. 전체적인 방향성을 의논해 본 결과, 우리는 웹이라는 특성을 살릴 수 있게 다양한 DB를 구축하고 각 회원 별로 자신만의 재생목록을 관리하며 음악을 감상할 수 있게 하고, 간단한 Social Network Service를 구현하여 친구의 최근 감상 목록을 조회할 수 있다던지, 자신이 좋아하는 그룹의 활동 내역을 볼 수 있게 하는 부분을 구현하였다. 또한 한 웹 페이지에서 모든 기능을 수행하면서도 불편함이 없도록 설계하였다. 앨범을 누르면 앨범에 수록된 곡 목록이 새로운 창으로 뜨지 않고 바로 그 창에서 팝업 애니메이션이 뜨게 하고, 재생 중에 페이지 전환 요청이 일어나도 음악 재생이 끊기지 않도록 비동기적으로 페이지 내용을 가져오도록 하는 등 페이지 전환의 불편함을 없애고 깔끔하게 기능들을 구현하였다. 그리고 웹의 특성 상 여러 다른 조건 하에 웹에 접속하게 된다. 이러한 특성 상 우리는 반응성 웹을 구현하여 각종 해상도와 브라우저(다만 Flash Plugin은 지원하여야 한다)에 맞춰 최적화된 페이지를 제공하게 하였다. 물론 기본적인 음악 Streaming 감상이라는 측면에도 충실하게 Player가 가져야 할 기본 기능들도 충실히 구현하였다.

***Login 화면***

# 사용된 기술 및 코드

먼저, 워낙 많은 코드가 사용되어 보고서에 모든 부분을 담을 수 없어서 코드 전체를 제출하고 코드마다 어떤 역할을 하는지 약간의 주석을 달아서 따로 첨부하였다. 여기서는 어떠한 기술들이 사용되었고 그 구현의 결과가 어떤지를 설명하겠다.

**Programing Flow Chart**

우리 프로젝트는 위의 Flow Chart에 나와있듯이 크게 Web Design, Client Programing, Server Programing세 부분으로 나뉜다. 이 세 부분에서 사용된 기술들에 대한 설명과 약간의 코드, 그리고 그 기술들을 사용함으로써 어떠한 결과가 나왔는지를 기술하였다.

Web Design Part에서는 HTML5, CSS(Bootstrap)을 사용하여 개발을 진행하였다. HTML 5는 Hyper Text Markup Language로 웹 사이트의 구조를 담고 있다. Android 프로그래밍의 XML 부분과 유사하다고 보면 된다. (둘 다 Markup Language이기 때문이다.)

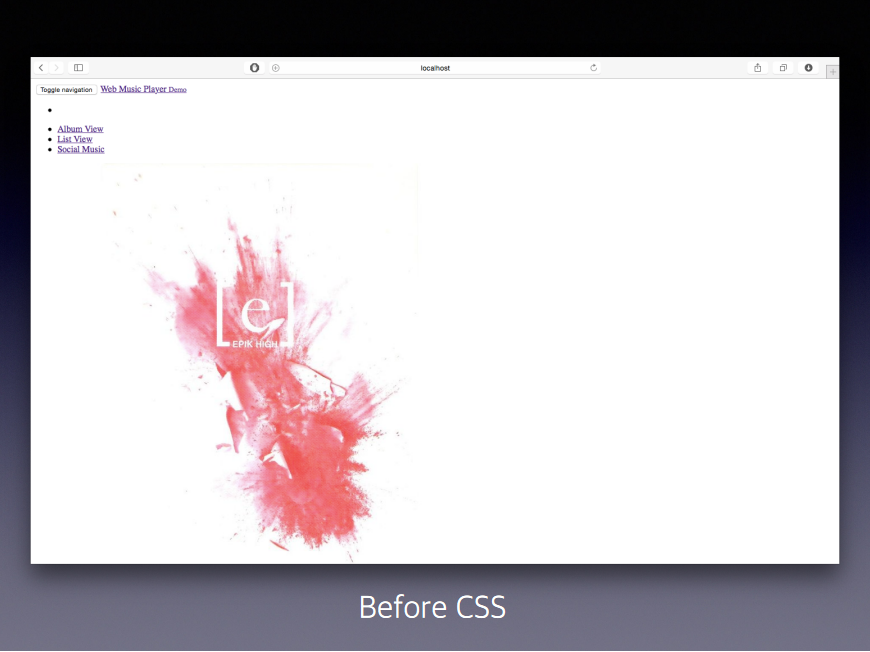


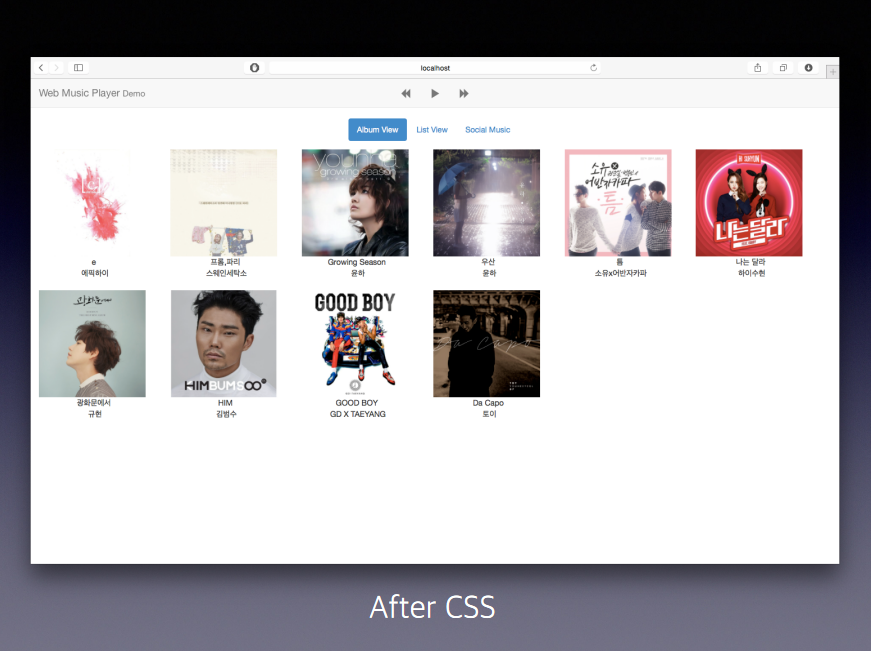
**사용된 HTML 코드 중 일부. 이와 같이 태그들을 정의한다.**

CSS는 웹 페이지의 각 부분의 Style을 담고 있는 부분이다. 각 태그마다의 폰트, 색, 크기 등의 웹 페이지의 테마 정보가 기술되어있다. 또한 우리는 개발의 편의성을 위하여 Bootstrap이라는 CSS, JavaScript 개발도구를 사용하였다. 이는 버튼, 아이콘, 메뉴 바 등의 다양한 기능의 디자인 템플릿을 Open-Source로 제공하여 디자이너 없이도 충분히 깔끔한 디자인의 웹을 구현할 수 있었다.

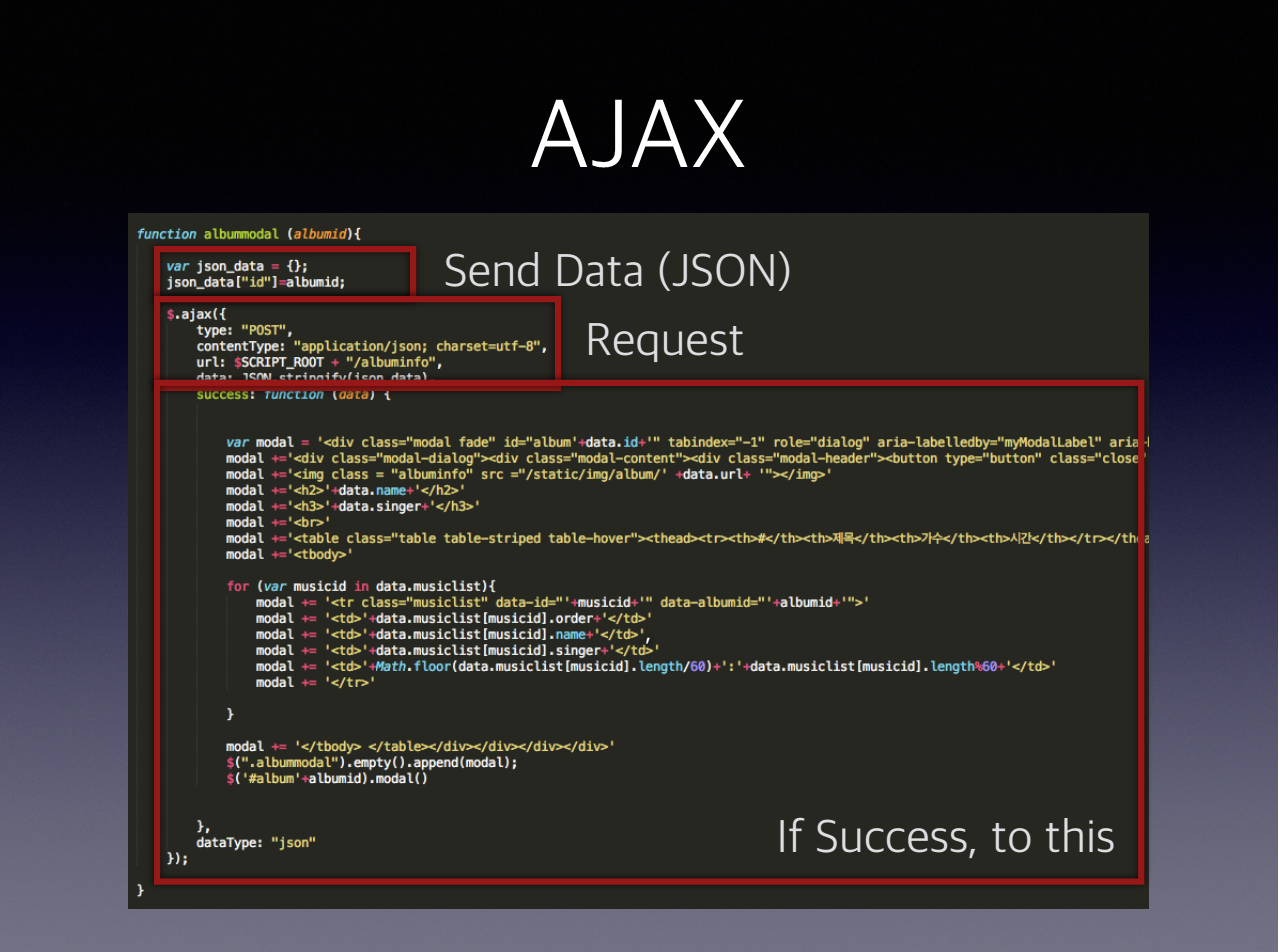


**CSS를 태그에 Import시키는 코드이다.**

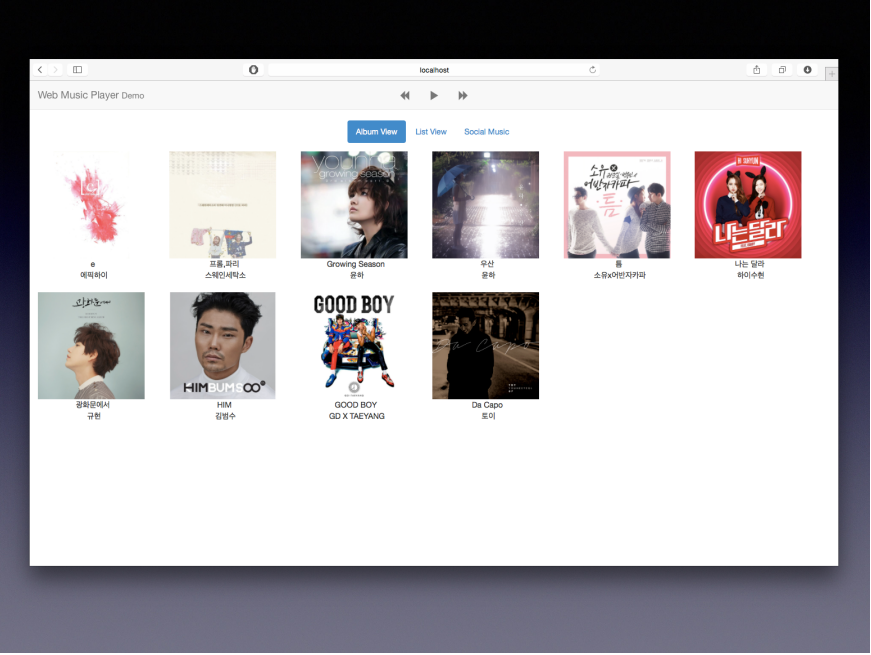




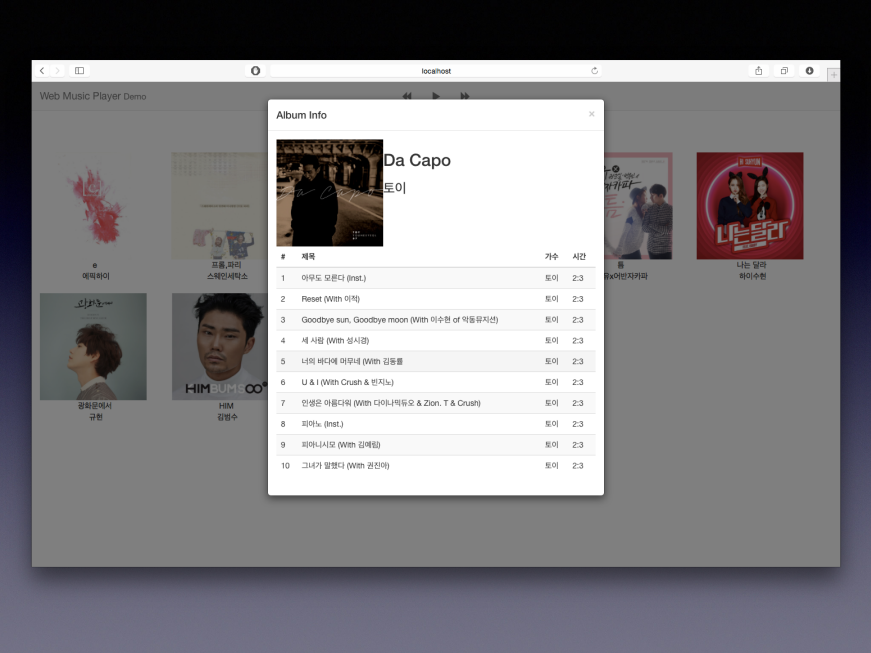
Client Programing Part에서는 JavaScript와 JQuery, Flash를 사용하여 개발을 진행하였다. JavaScript는 능동적으로 웹 페이지를 다룰 수 있게 하기 위해 필요한 것으로, HTML 태그들을 유동적으로 제어할 수 있게 해주는 언어이다. Client에서 처리할 연산들을 담당하고, 이를 다시 HTML로 전송해주어 웹 페이지에 표시되는 정보를 전달해준다. jQuery는 JavaScript를 사용할 때 모든 조건의 웹에서 잘 작동할 수 있게 호환성을 담당하는 라이브러리이다. 위에 개요에서 얘기했듯이 우리는 페이지 변환 없이 비동기적으로 서버와 통신하는 AJAX라는 웹 개발 기법을 사용하는데 이 AJAX 또한 jQuery를 이용하여 처리하게 된다. 마지막으로 Flash는 음성을 전달하고 처리하는데 적합한 Client Plugin이다. 우리는 Flash에서 지원하는 프로토콜인 rtmp media stream을 받아서 재생하기 위해 ActionScript를 이용해 Flash 부분을 구현하였다. 다음 사진들은 AJAX로 구현한 기능을 사진으로 찍어 본 것이다. 앨범 표지를 클릭할 때, 페이지가 전환되는 것이 아니라 앨범 팝업이 뜨게 함으로서 음악이 끊기지 않게 함과 동시에 페이지 로딩의 번거로움을 없앴다.



**AJAX를 이용한 비동기성 페이지 로딩 수행 코드**

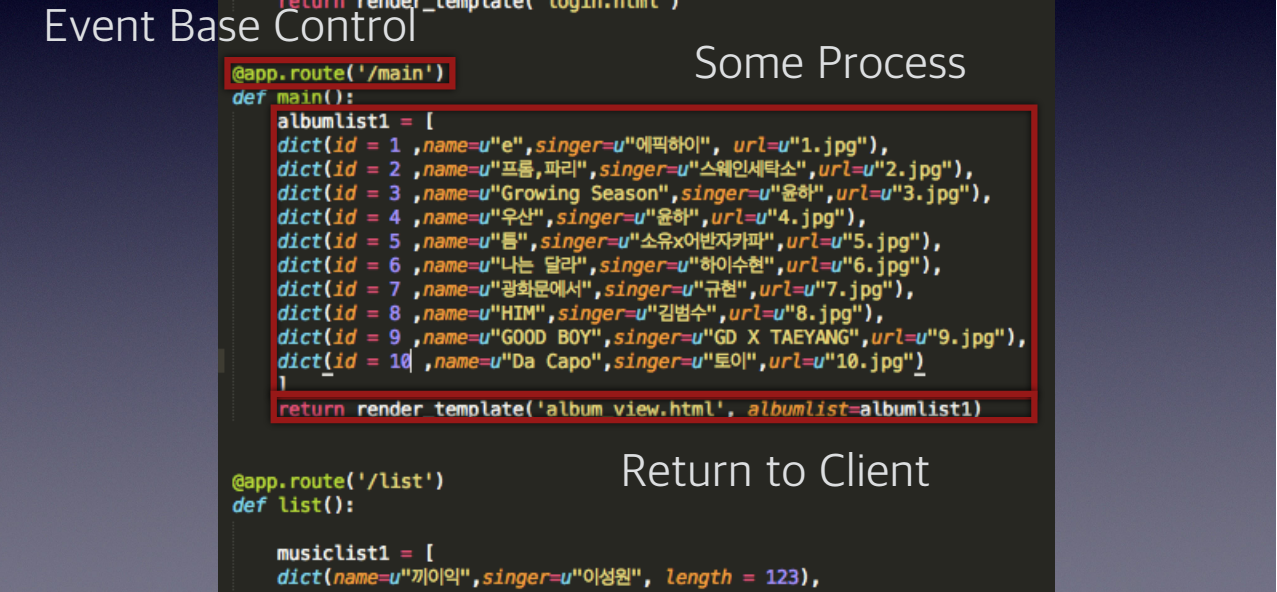


**앨범 표지를 누르면 아래 사진과 같이 팝업 형식으로 애니메이션이 뜨게 된다.**



**AJAX를 이용함으로써 비동기성 페이지 로딩이 가능**

마지막으로 Server Programing Part에서는 Flask, Jinja2, SQLAlchemy, gstreamer, nginx-rtmp 등을 사용하였다. Server란 Client에서 요청이 들어왔을 때, 그 요청에 대한 응답을 해주는 부분이기에 Flask라는 웹 이벤트를 처리하는 프레임워크를 사용하였다. Flask에서 요청에 따른 응답이 나오면 이를 Jinja2라는 템플릿 엔진을 사용하여 Client로 전달 할 정보를 HTML 규격으로 변환해 준다.



**Flask를 이용한 Event 처리 코드**

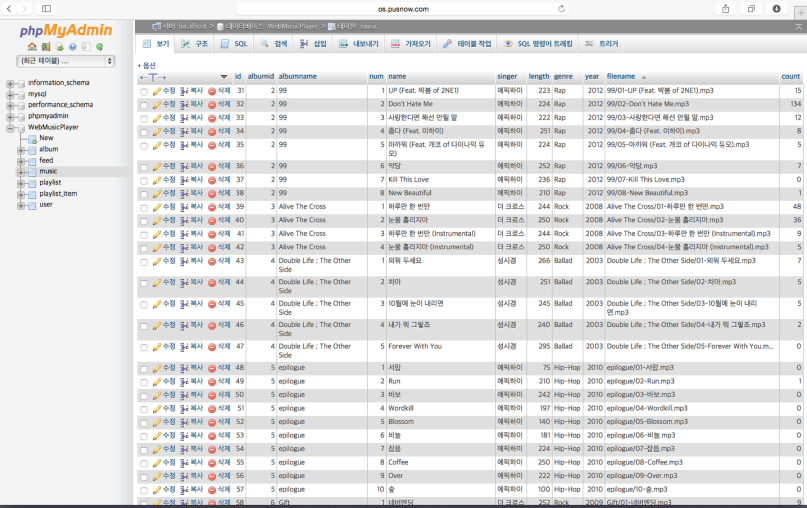


**Jinja2를 Call하여 HTML 코드로 변환**



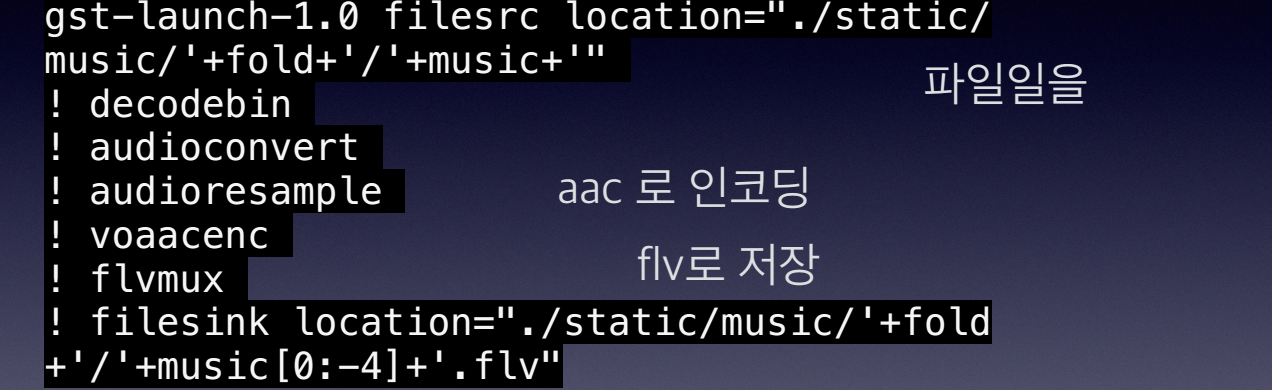
**Jinja2 코드. 이러한 방식으로 Client에 전달할 정보를 가공한다.**

Server Part에서는 전체 DB도 관리하게 되는데 우리는 MariaDB라는 관리 시스템과 SQLAlchemy라는 모듈을 이용하여 DB를 관리하였다.

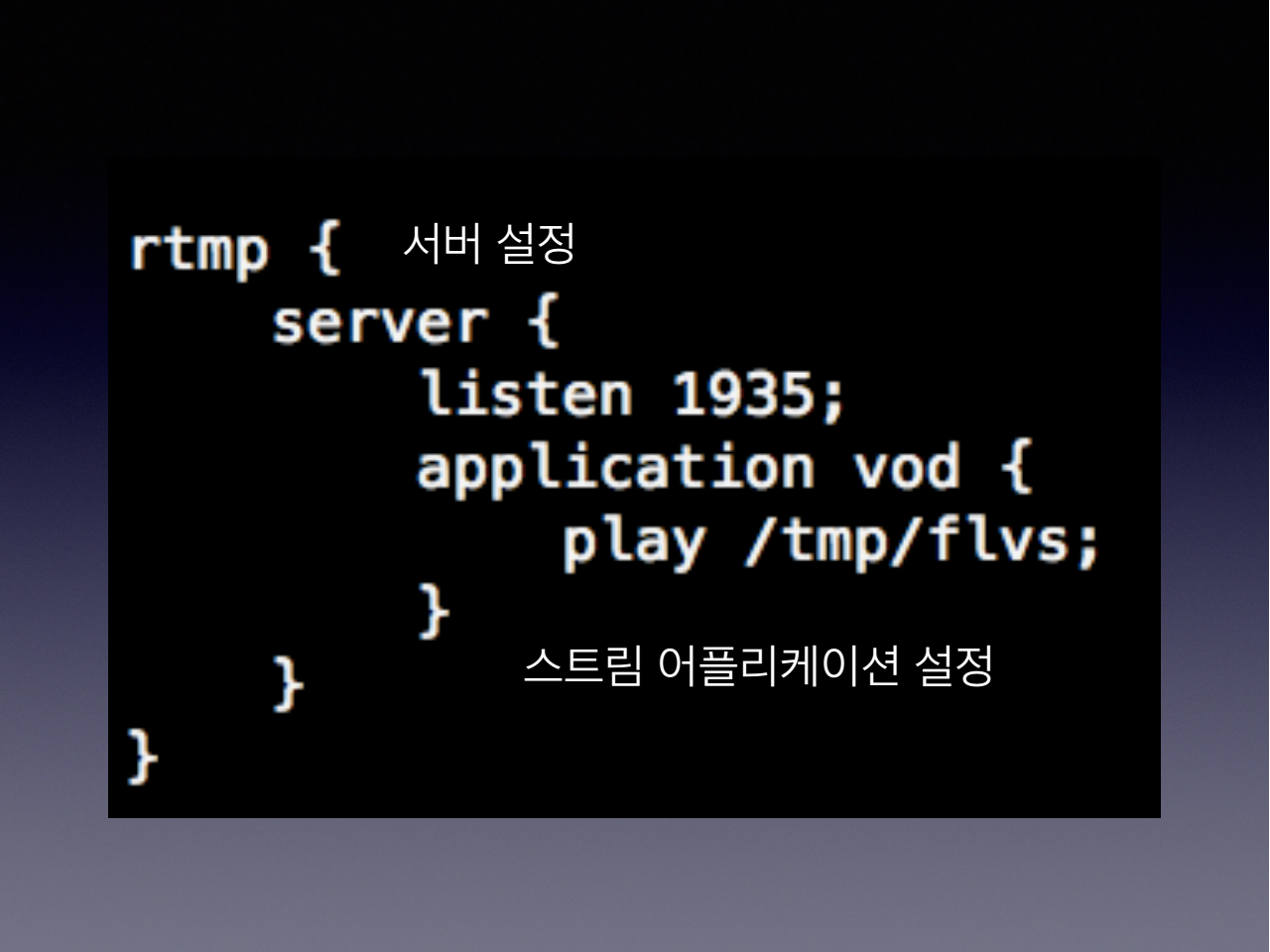


**음악 DB 관리 페이지**

또한 제일 중요한 음악 Streaming 부분은 gstreamer라는 멀티미디어 프레임워크를 이용하여 미디어 정보를 추출하여 rtmp 프로토콜에서 지원하는 형식인 flv 컨테이너에 담아주게 된다. nginx-rtmp 서버를 구성하여 인코딩을 통하여 제작된 Media Stream을 Client로 전달해주어 Flash를 이용하여 재생하게 된다.



**Mp3 파일을 flv로 인코딩하는 Gstreamer 코드**



**Rtmp 서버 설정 코드**

# 프로젝트 시연 및 설정 방법

OS.pusnow.com에서 직접 프로젝트를 시연해볼 수 있다.

관리자 ID : [admin@admin.com](mailto:admin@admin.com)

비밀번호 : iloveos

|  |
| --- |
| 또한, 직접 서버를 설정하고 돌리는 법도 간략하게 기술한다. |
| \* 이 세팅은 Ubuntu Server 14.04 LTS 버전에서 진행하였습니다.  1. nginx-rtmp-module이 활성화된 nginx를 설치합니다. (참고 : https://github.com/arut/nginx-rtmp-module/wiki/Installing-on-Ubuntu-using-PPAs)  2. python과 pip를 설치합니다. apt-get 사용  3. pip 를 이용해 다음 모듈을 설치합니다. flask, sqlalchemy  4. 필요한 코덱을 참고하여 gstreamer와 플러그인들을 설치합니다. apt-get 사용 gstreamer-tool, gstreamer-base, gstreamer-good 등 (참고 : http://gstreamer.freedesktop.org/documentation/plugins.html )  5. uwsgi를 설치합니다. pip 사용  6. mariaDB를 설치합니다. apt-get 사용  7. 스키마를 DB에 import합니다. WebMusicPlayer.sql  8. DB 계정을 만들어 권한을 주고, database.py 코드를 수정하여 DB 연결설정을 합니다.  9. 소스코드를 서버에 올립니다.  10. 폴더이름을 앨범 명으로, 그 앨범 안에 mp3파일을 저장합니다.  11. 첨부한 엑셀 파일 규격에 맞추어 파일을 작성합니다.  12. MySQL ODBC 를 이용해 album과 music DB에 저장합니다. (Struct를 수정해야 합니다. Primary Key 추가)  13. makeflv.py 파일이 있는 곳에 들어가 python makeflv.py 을 입력해 flv 파일로 인코딩합니다.  14. DB에 User를 수동으로 추가합니다. 비밀번호는 md5로 암호화 합니다.  15. DB에 Playlist를 수동으로 추가합니다. userid 는 14번에서 만든 User id, 각 id 마다 첫 번째 playlist는 좋아하는 음악입니다.  16. 서버에 /tmp 폴더가 있나 확인하고 이 폴더 안에 flv 폴더를 만듭니다. 권한 설정 또한 웹서버가 접근 가능 하게 777권한을 줍니다.  17. 웹서버와 rtmp 서버 세팅을 합니다. nginx.conf, site-available/default 파일을 첨부한 파일을 수정합니다.  17. runserver.py가 있는 폴더에서 다음 명령을 입력합니다. uwsgi -s /tmp/uwsgi.sock --module WebMusicPlayer --callable app&  18. /tmp 폴더에 들어가 uwsgi.sock 파일의 권한을 확인하여 웹서버가 접근 가능하게 합니다. |