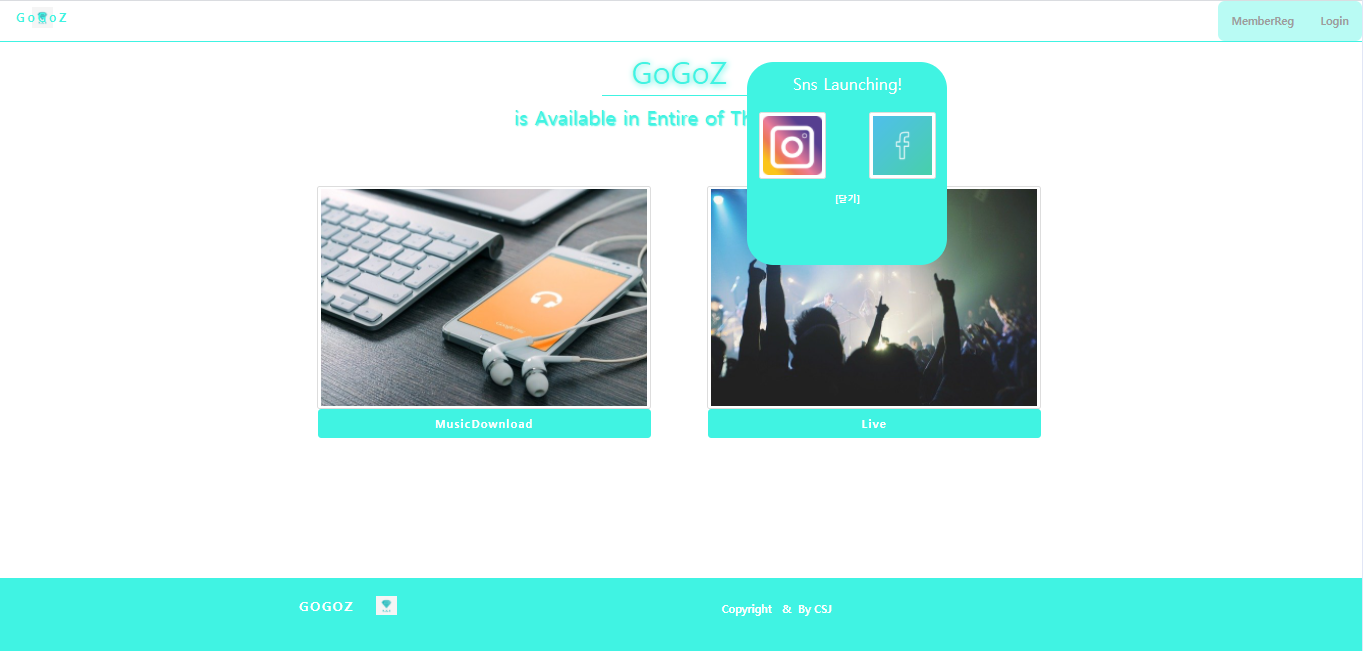
**GoGoZ 뮤직 스트리밍 플랫폼**

개발자: 천세진(Brant)  
기획/디자인: 천세진(Brant)  
프로젝트 주소:   
완료 여부: 진행중

**목차**

1. 사용 기술 소개-----------------------------------------------------------------------------------
2. 페이지 구성--------------------------------------------------------------------------------------
3. 특이점--------------------------------------------------------------------------------------------
4. 각 기능별로 사용되는 DB 테이블--------------------------------------------------------------
5. JS파일 기능별 설명------------------------------------------------------------------------------
6. 시큐어 코딩 적용--------------------------------------------------------------------------------
7. 기술 사용(작업) 기록 ---------------------------------------------------------------------------

* **사용 기술 소개**  
    
  통합개발환경(IDE)은 이클립스, 데이터베이스 툴은 Mysql(Workbench)을 사용했고,  
  **Server단(B/E)은 JAVA**를, **Client단(F/E)은 HTML5 기반**으로 설계하였습니다.  
    
  더 자세하게 풀어 소개하면, 서버의 기능을 담당하는 부분은   
  스프링 프레임워크(Spring MVC)를 사용하였으며,  
  뷰(VIEW)는 HTML5와 CSS3, JS/Jquery를 사용하여 구축하였습니다.  
    
  하지만, 크기가 무거운 Jquery보다 바닐라JS(순수자바스크립트)가  
  자리를 잡아가는 추세인 만큼, Jquery를 사용하기보단  
  온전히 순수 자바스크립트를 사용함으로 웹 사이트의   
  동적인 부분을 구현해 내었습니다.  
    
  모바일 기기의 수요량이 필수라고 불릴 정도로   
  증가하고있는 만큼 반응형기능을 추가함으로 디바이스의 크기에 따라 웹 사이트의  
  크기 또한 조정되도록 가능하게끔 제작하였습니다.  
    
  또한, 대부분의 하드웨어와 소프트웨어가 지능화 및 자동화되어가는 추세임을  
  참고하여 입력을 담당하는 하드웨어의 통제없이 해당 플랫폼을 이용할 수 있도록  
  음성인식(Speech-Recognition API)기능을 추가하여 보았습니다.
* **페이지 구성  
  (회원, 게시판 관련 기능 URL 요청은 \*.do 방식으로 설정하였음.)  
  (음악 관련 기능 URL 요청은 \*.music 방식으로 설정하였음.)  
  (Socket 관련 기능 URL 요청은 \*.bo 방식으로 설정하였음.)**

1. 초기화면:   
   View: <http://localhost:8080/>  
     
   ServerRequest:(/index.do)
2. 회원가입(JSON)

View: <http://localhost:8080/MemberRegister/MemberRegister.jsp>  
  
ServerRequest:("/MemberReg/MemberReg.do")

1. 로그인(JSON)  
   View:   
   <http://localhost:8080/MemberLogin/login.jsp>  
     
   ServerRequest:("/MemberLog/MemberLog.do")
2. 회원탈퇴:  
   View:  
   <http://localhost:8080/MemberDelete/delete.jsp>  
     
   ServerRequest:(/MemberDel/MemberDel.do)
3. 로그아웃  
   ServerRequest:(/MemberLogout.do)
4. 회원 수정  
   View:  
   <http://localhost:8080/MemberUpdate/update.jsp>  
     
   ServerRequest:("/MemberUpdate/MemberUpdate.do")
5. 질문 게시판  
   View:  
   <http://localhost:8080/Board/BoardForm.jsp>  
     
   ServerRequest:("/InquiryBoard/Inquiry.do")
6. 관리자 기능(HandlerInterceptorAdapter)   
   
7. 채널 생성  
   View:  
   <http://localhost:8080/Chanel/ChanelCreateView.jsp>  
     
   ServerRequest:("/function/Chanel/ChanelCreate.bo")
8. 음악 다운로드 리스트  
   View:  
   <http://localhost:8080/Music/MusicList.jsp>

ServerRequest:("/Music/MusicList.music")

1. RTC 접속(컨트롤러 클래스 도달 보류)  
   View:  
   <http://localhost:8080/Function/function.jsp>

ServerRequest:("/Function/function.music")



* **특이점**

사람의 자체적인 모션이 필요없이 하드웨어를 동작시킬 수 있도록   
음성인식(SpeechRecognition API) 기능을 추가하여 보았습니다.

단말기를 통한  
유저의 명령 인식

DB에서 가장 근접한  
 데이터를 찾아 페이지  
 이동

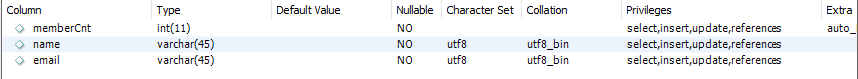
회원가입

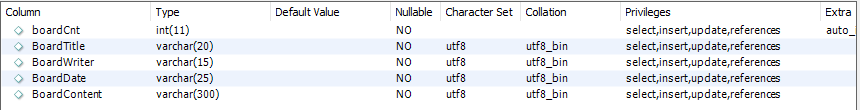
실행 종료

로그인

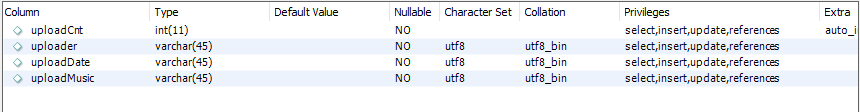
어플리케이션이 실행되면 클라이언트 측에서 유저의 명령을 받는 기능을 실행합니다.  
소프트웨어가 유저의 명령을 정상적으로 인식하게 될 경우, 유저로부터 받은 데이터와   
가장 근접한 의미를 가진 동작을 데이터베이스에서 찾은 후, 어플리케이션 내부에서  
실행하도록 합니다.

* **각 기능별로 사용되는 DB 테이블 (사용 스키마 이름 => prototype)**

회원정보 관리 테이블(**member**)  


게시판 관리 테이블(**board**)  


업로드 된 음악 관리 테이블(**UploadedFile**)



* **JS파일 기능별 설명**

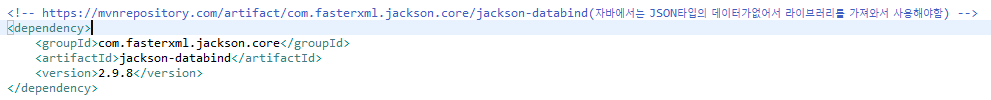
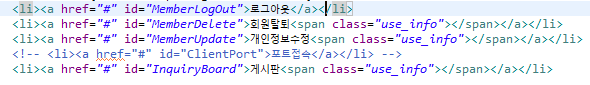
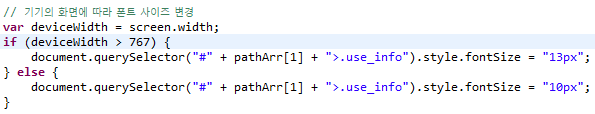
|  |  |
| --- | --- |
| **CustomFrameWork/style&js/js/Music/MusicJS.js** | **관리자의 음악 업로드,  사용자의 음악다운로드  기능 담당** |
| **CustomFrameWork/style&js/js/RTC/RTC.js** | **스트리밍 웹 캠 동작 담당** |
| **CustomFrameWork/style&js/js/RTC/Socket.js** | **채널 생성 담당** |
| **CustomFrameWork/style&js/js/SpeechRecognition/speech.js** | **음성 인식 담당** |
| **CustomFrameWork/style&js/js/CustomEffect.js** | **CapsLock을 누르거나, 게시글 작성창을  2번 누를 때  발생하는 이벤트 담당** |
| **CustomFrameWork/style&js/js/CustomJs.js** | **기본적인 회원 기능 (로그인,회원가입, 회원 수정,회원탈퇴, 로그아웃 등등)담당** |
| **CustomFrameWork/style&js/js/CustomJs.js** | **인덱스 페이지에  출력되는 팝업 Ajax 담당** |
| **CustomFrameWork/style&js/js/Popup.js** | **팝업에 붙어있는 SNS그림을 누르면  해당 SNS주소로 이동** |
|  |  |

* **시큐어 코딩 적용**



1. 로그인 5회 실패 시 24시간 서비스 이용 불가(스프링 시큐리티 적용)  
   

* **기술 사용(작업) 기록**

1. 기존 Spring내부에서 적용되어 있는 기본적인 방식으로 뷰를 출력해주었던  
   **Inner View Resolver**(String, Model And View, Object 설정에 따른 View단 출력)  
   에서 Ajax을 사용해 POST방식으로 전달될 JSON형식의 데이터와  
   @ResponseBody를 이용한 데이터 출력 및 View페이징 방식으로   
   전환해주는 **Message Converter**를 적용하였습니다.  
   여기서, 컨트롤러 내부에서 **@Responsebody가 적용된 함수의 리턴 값은   
   뷰를 출력해주는 것이 아닌 리턴 된 있는 그대로의 데이터를 출력**하게 됩니다.  
   **@ResponseBody가 적용된 함수에 설정한 매개변수에 @RequestBody를   
   적용하면 그 매개변수에는 클라이언트에서 전송한 data가 담아집니다.**   
   함으로서, 자바 객체를 JSON으로 변환해서 보여주는 뷰 구현 클래스인  
   **MappingJackson2JsonView** 클래스를 사용해서 뷰를 그려주도록 합니다.  
   
2. **2021/02/27** 부터  
   기존 **채널 접속과 관련한 기능(Chanel)**은 DAO(Data Access Object)와  
   Service의 구현없이 독립적인 Generator 클래스 형식으로 컨트롤러에   
   주입해서 바로 적용을 시킬 수 있도록 만들었었지만,   
   코드의 업데이트와 로직 구분에 어려움이 있어 회원 기능과   
   비슷하게 DAO와 Service를 구현함으로 기능 분석에 소요되는 시간을   
   최소화 시켰습니다.  
   **기존ex)** URL\_Mapping - > Controller < - > ChanelGenorator.java
3. **2021/03/07** 부터  
   메뉴에서 선택한 텍스트가 현재 열려있는 페이지 일 경우  
   해당 텍스트에 ‘(사용중)’ 띄우기(**JS에서 window.location객체** 사용)  
     
     
   **window.location 객체 값** 정보)  
     
     
   **결과)**  
   
4. **2021/3/20** 부로 기기의 크기에 따라 스타일(CSS)과 JS가 적용되도록 업데이트.  
   (JS에 **screen 함수** 사용)  
     
   단, IE8이하의 브라우저일 경우 미 적용.(지원되지 않도록 구현함.)