**웹 프로그래밍 팀 프로젝트**

**최종 보고서**

컴퓨터 공학과

2017112129 이정욱

2017112090 김동욱

2017112125 조현준

송양의 교수님



목차

[**프로젝트 개요** 3](#_Toc58390726)

[**주제** 3](#_Toc58390727)

[**기획 의도** 3](#_Toc58390728)

[**파일 구조 및 트리** 4](#_Toc58390729)

[**로그인 스크린** 5](#_Toc58390730)

[**로그인 화면** 5](#_Toc58390731)

[**프론트엔드** 6](#_Toc58390732)

[**백엔드** 6](#_Toc58390733)

[**회원가입 화면** 7](#_Toc58390734)

[**프론트엔드** 7](#_Toc58390735)

[**백엔드** 7](#_Toc58390736)

[**아이디 / 비밀번호 찾기 화면** 7](#_Toc58390737)

[**프론트엔드** 7](#_Toc58390738)

[**백엔드** 7](#_Toc58390739)

[**캘린더 스크린** 8](#_Toc58390740)

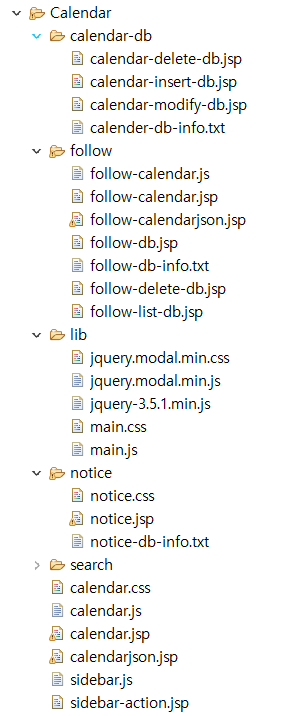
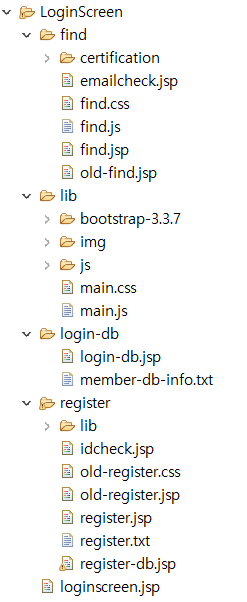
# **프로젝트 개요**

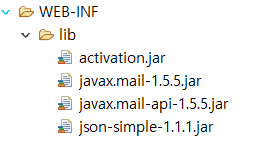
## **주제**

## **기획 의도**

# **파일 구조 및 트리**

프로젝트 전체 파일은 다음과 같은 구조이다.





LoginScreen 폴더에는 로그인과 회원가입, ID/PW 찾기, 메일 인증 등을 담당하는 파일들이 위치해있다. 각 하위 폴더를 살펴보면 find폴더는 ID/PW 찾기 페이지와 메일인증 기능을 담당하는 폴더이다. lib폴더는 로그인 스크린의 라이브러리, 이미지, js, css 파일이 위치해 있다. login-db는 로그인 기능을 담당하는 폴더이다. registe는 회원가입 기능을 담당하는 폴더이다.

Calendar 폴더에는 실제 캘린더의 기능이 구현된 폴더이다. calendar-db는 일정 DB의 추가, 삭제, 수정 기능을 담당한다. follow 폴더는 친구 팔로우, 팔로우 삭제, 팔로우 한 사람의 캘린더 열람 기능을 담당한다. lib 폴더는 라이브러리, js , css 파일이 위치하고있다. notice는 누가 나를 팔로우 했는지 알려주는 기능을 담당한다. search 폴더는 일정 검색 기능을 담당한다.

WEB-INF\lib 폴더에는 메일 인증과 json을 다룰 때 필요한 jar 파일들이 위치해있다.

# **로그인 스크린**

## **로그인 화면**



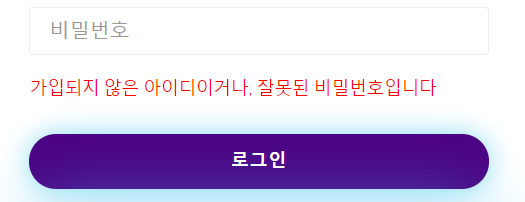
loginscreen.jsp 실행시 출력되는 화면이다.

### **프론트엔드**

프론트엔드 라이브러리인 Bootstrap, Owl Carousel, Typed.js를 이용하여 로그인 화면의 웹 디자인을 구성하였다. 전체적인 디자인은 Bootstrap의 그리드 시스템과 버튼 클래스를 활용하여 디자인했고 main.css로 저장하였다. Web calendar 밑에 설명은 Typed.js를 이용하여 글쓰는 효과를 적용시켰고 하단에 사용 라이브러리들의 로고를 Owl Carousel을 이용해 슬라이드 형태로 보여준다. Typed.js와 Owl Carousel을 이용한 자바스크립트 코드는 main.js로 저장하였다.

### **백엔드**

아이디와 비밀번호 입력부분은 <form> 태그로 되어있다. 로그인 버튼은 submit 타입이고 회원가입 버튼은 button타입이다. 로그인 버튼을 누르면 아이디는 id이름의 파라미터로, 비밀번호는 pw 파라미터로 login-db.jsp 에 전달된다. 로그인에 성공하면 calendar.jsp로 redirect 시키지만, 로그인에 실패하면 실패 메시지를 다시 loginscreen.jsp로 전달하여 아래와 같이 출력한다.



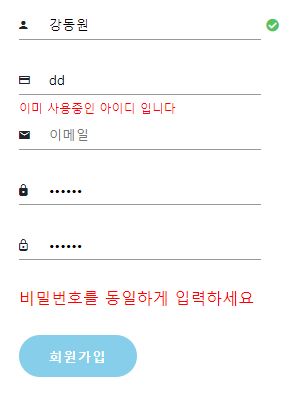
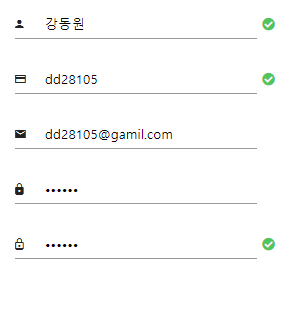
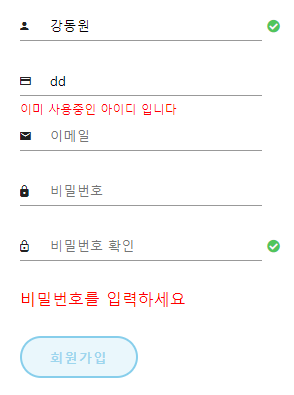
## **회원가입 화면**



loginscreen.jsp에서 회원가입 버튼 클릭시 이동하는 페이지이다.

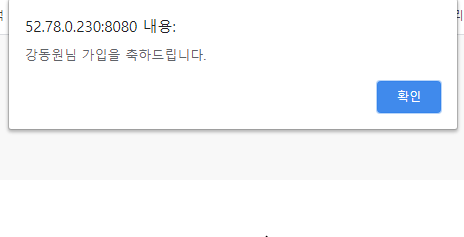
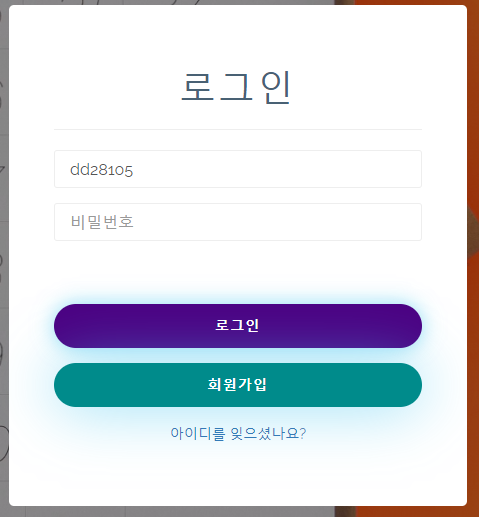
### **프론트엔드**

### **백엔드**

회원가입 정보입력은 구체적인 제한은 두지 않았으며 입력된 정보의 유효성만을 확인한다. 모든 칸이 입력되었는지, 아이디가 중복되는지, 비밀번호를 두 칸 동일하게 입력했는지를 확인한다. 검사는 실시간으로 반영되며, 모두 통과해야만 회원가입 버튼이 활성화된다.

<form>태그는 이름, 아이디, 이메일, 비밀번호 등의 입력칸과 회원가입 버튼으로 구성되며 <form>의 onsubmit 속성, <input>의 onkeyup 속성을 이용하여 각각 제출할 때, 정보를 입력할 때 자바스크립트 함수를 호출하며, 유효성을 검사한다. 유호성 검사를 통과하여 제출이 되면 name, id, email, pw 파라미터가 register-db.jsp로 전달되며 member

테이블에 새로운 정보로 추가한다.

정보 추가가 완료되면 가입축하 알림을 띄우고 id 파라미터와 함께 로그인 스크린으로 전달한다. 로그인스크린으로 돌아오면 아이디 입력부분이 가입한 아이디로 설정되어있다.

## **아이디 / 비밀번호 찾기 화면**

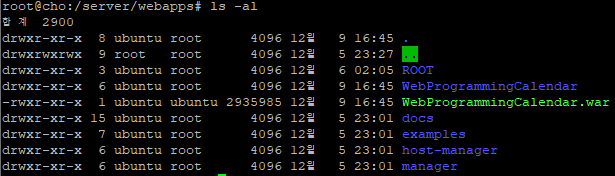
### **프론트엔드**

### **백엔드**

# **캘린더 스크린**

# **배포**

해당 프로젝트를 배포하기 위해 AWS EC2를 이용하였다. AWS EC2에서 Ubuntu 18.04 인스턴스를 생성하여 Tomcat 및 mysql 환경을 구축하고, 인스턴스의 외부 접속 IP에서 일부 포트를 외부에서 접속 가능하게 설정했다.



* webapps 폴더에 있는 프로젝트의 war 파일



* conf/server.xml의 일부