

웹 프로그래밍 Term Project
최종 보고서

제출일자	2019.06.13
분 반	02
학 번	201502085
이 름	이규봉

부트스트랩 4를 이용한 반응형 채팅 웹 페이지

분반 : 02반

학번 : 201502085

이름 : 이 규 봉

1. 과제의 개요

1) 본 과제의 주제

: 주제는 채팅 웹 페이지 구현입니다. 보다 구체적으로는 데이터 베이스에 채팅 데이터를 저장하고 fetch 해 오는 식으로 채팅 웹 페이지 및 관련 서비스를 구현하는 것 입니다.

2) 본 과제에서 창의적이라고 생각하는 부분

1: 프론트를 부트스트랩을 사용하여, 반응형 웹페이지로 구현하였습니다. 따라서, 데스크톱 뿐 만 아니라 스마트폰이나 태블릿 등의 디바이스에서도 용이하게 사용할 수 있습니다. 채팅 서비스는 반드시 스마트폰으로도 용이하게 이용할 수 있어야 한다고 판단해, 설계 초기부터 부트스트랩을 사용해 제작했으며, 미디어 쿼리를 이용해 웹페이지들을 자연스러운 반응형 웹페이지로 구현하기 위해 신경썼습니다.

2: 백엔드에선 데이터베이스를 적극적으로 활용하였습니다. 데이터베이스에 프로필 사진의 경로를 저장해 놓고, 웹 페이지에서 파일을 찾아와 띄우는 등이 데이터베이스를 적극적으로 활용한 예입니다.

3: Ajax를 사용해 채팅 웹페이지가 리로드 되지 않고도 채팅을 쉽게 일어날 수 있게 하였습니다. 채팅방마다 저마다의 table로 구현되어 있고, 각자 Auto increment 값을 Primary Key로 하기 때문에, MySQL 내부에서 비교적 빠른 성능을 가질 것으로 보입니다. Ajax로 데이터를 fetch 해 올 때도, 각 클라이언트의 브라우저에 업데이트 되지 않은 데이터만을 조회해 가져옴으로써, 불필요한 서버 부하를 최소화 시키려 했습니다.

4: 웹 소켓을 사용하지 않고, ajax로 DB에 채팅 데이터를 저장하는 연산과, 가져오는 연산만으로 채팅 프로그램을 만들려 했습니다. 웹 소켓을 사용해 구현해도 채팅 데이터를 모두 저장해 놓으려면 DB가 필요하기 때문에, 차라리 웹 소켓을 사용하지 않고 채팅 프로그램을 구현하는 것이 더 효율적이지 않을까 생각하고 구현했습니다. 다만, 이런 방식의 구현은 각 클라이언트들이 ajax를 일정 시간마다 계속해서 호출하기 때문에, 접속자가 많아지면 서버의 부하가 커질 수 있을 것 같습니다. 접속자가 많아져 서버의 부하를 줄여야 하는 상황이 된다면, 1초마다 MessageFetchAction.php를 실행하는 콜백 함수를 제거하고, HTML5 API의 웹 소켓을 사용해 채팅방 table에 새 레코드가 들어오면 해당 채팅방 사용자들의 브라우저들에 Server Sent Event를 발생시켜, 이 때 만 MessageFetchAction.php를 실행하도록 구조를 변경하여야 할 것 같습니다.

2. 기능별 디자인

※ 저번 보고서를 제출한 이후, 구현되어 있지 않던 채팅방 나가기, 다른 유저 초대 기능을 추가했으며, 일부 컴포넌트들은 색상 및 스타일을 변경하였습니다.

(1) 인터페이스

인터페이스 화면은 전반적으로 bootstrap.css의 스타일을 적극적으로 활용해 디자인 했습니다. 디자인에 사용되는 클래스 명 들은 bootstrap 4.3.2 API 나 구글 검색, 인터넷 강의 등에서 검색해 사용했습니다. 모바일 환경에서의 디자인 및 인터페이스를 계속 고려하며 프로젝트를 진행했기 때문에, 스마트폰에서의 화면도 함께 기재했습니다.

로그인 화면 :

데스크톱에서의 로그인 화면



A desktop view of a login page. It features a blue header with a '로그인' (Login) button. The main content area has a light blue background with the title 'Login' and a subtitle '채팅 서비스를 이용하기 위해 로그인하세요.' (Please log in to use the chat service). Below this are two input fields: 'ID' with the placeholder 'ID를 입력하세요' (Enter ID) and 'PW' with the placeholder '비밀번호를 입력하세요' (Enter password). A blue '로그인' (Login) button is positioned below the input fields. At the bottom left, there is a 'Sign Up' link. The footer contains the copyright notice '© 2019 웹프로그래밍 이규봉 Term Project'.

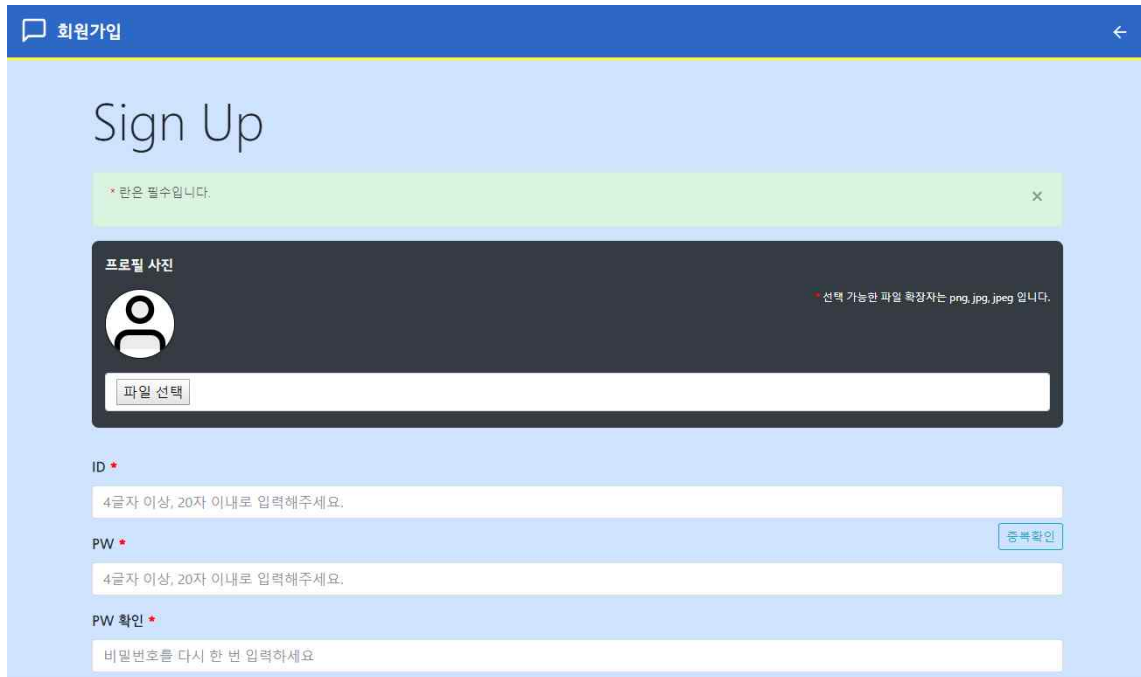
스마트폰에서의 로그인 화면



A mobile view of the same login page. The layout is adapted for a smaller screen, with the blue header and '로그인' button at the top. The 'Login' title and subtitle are centered. The input fields for 'ID' and 'PW' are stacked vertically. The blue '로그인' (Login) button is centered below the inputs. The 'Sign Up' link is at the bottom left. The footer with the copyright notice '© 2019 웹프로그래밍 이규봉 Term Project' is at the very bottom. The mobile interface includes standard Android navigation icons at the bottom.

회원가입 :

데스크톱에서의 로그인 화면



A desktop view of a 'Sign Up' form. The header is blue with a '회원가입' (Sign Up) button and a back arrow. The main title 'Sign Up' is in large grey font. Below it is a green error message box: '*란은 필수입니다.' (asterisk fields are required). The '프로필 사진' (Profile Photo) section has a dark grey background, a profile icon, and a file selection button labeled '파일 선택'. A note specifies '* 선택 가능한 파일 확장자는 png, jpg, jpeg 입니다.' (supported file extensions are png, jpg, jpeg). The form includes three input fields: 'ID' (4글자 이상, 20자 이내로 입력해주세요. - 4 characters or more, 20 characters or less), 'PW' (4글자 이상, 20자 이내로 입력해주세요. - 4 characters or more, 20 characters or less), and 'PW 확인' (비밀번호를 다시 한 번 입력하세요 - Enter password again). A '중복확인' (Check for duplicates) button is next to the password field.

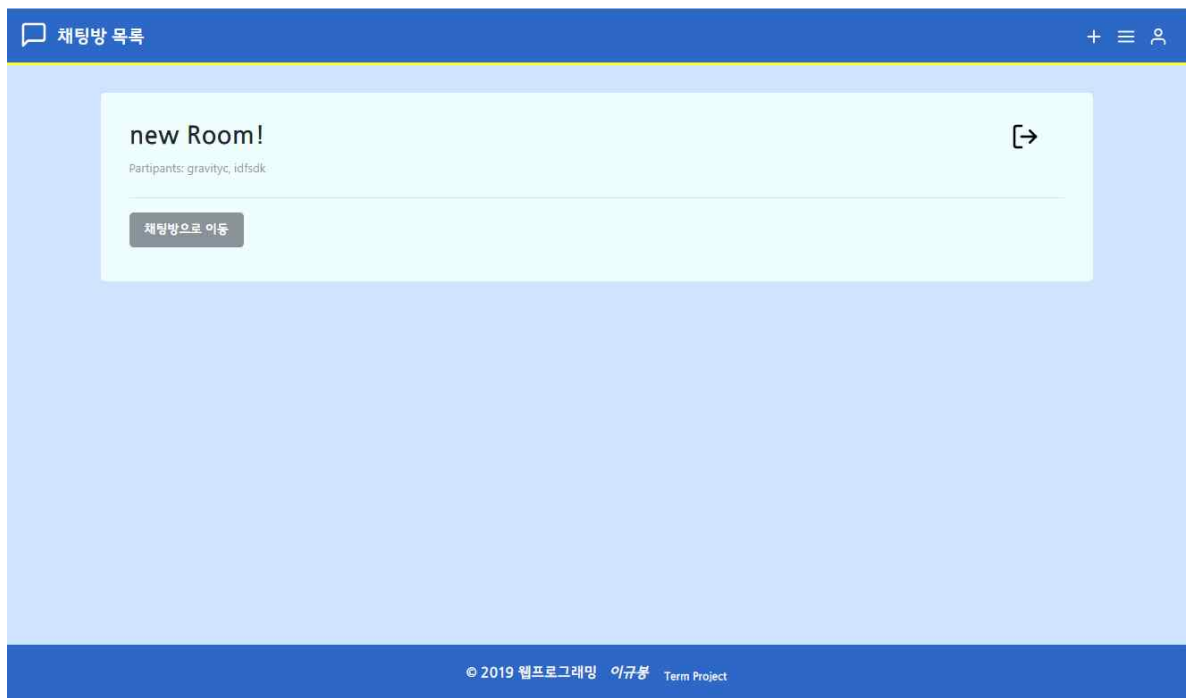
스마트폰에서의 회원가입 화면



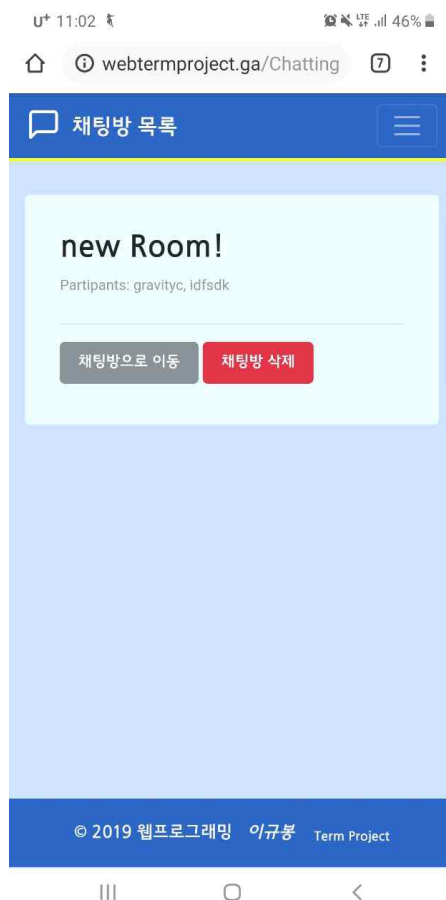
A mobile view of the 'Sign Up' form. The browser address bar shows 'webtermproject.ga/SignUp.t'. The form layout is adapted for a smaller screen. The '회원가입' header has a hamburger menu icon. The '프로필 사진' section is more compact. The 'ID' and 'PW' input fields are smaller. The '중복확인' button is more prominent. The bottom of the screen shows standard Android navigation icons.

채팅방 선택 화면:

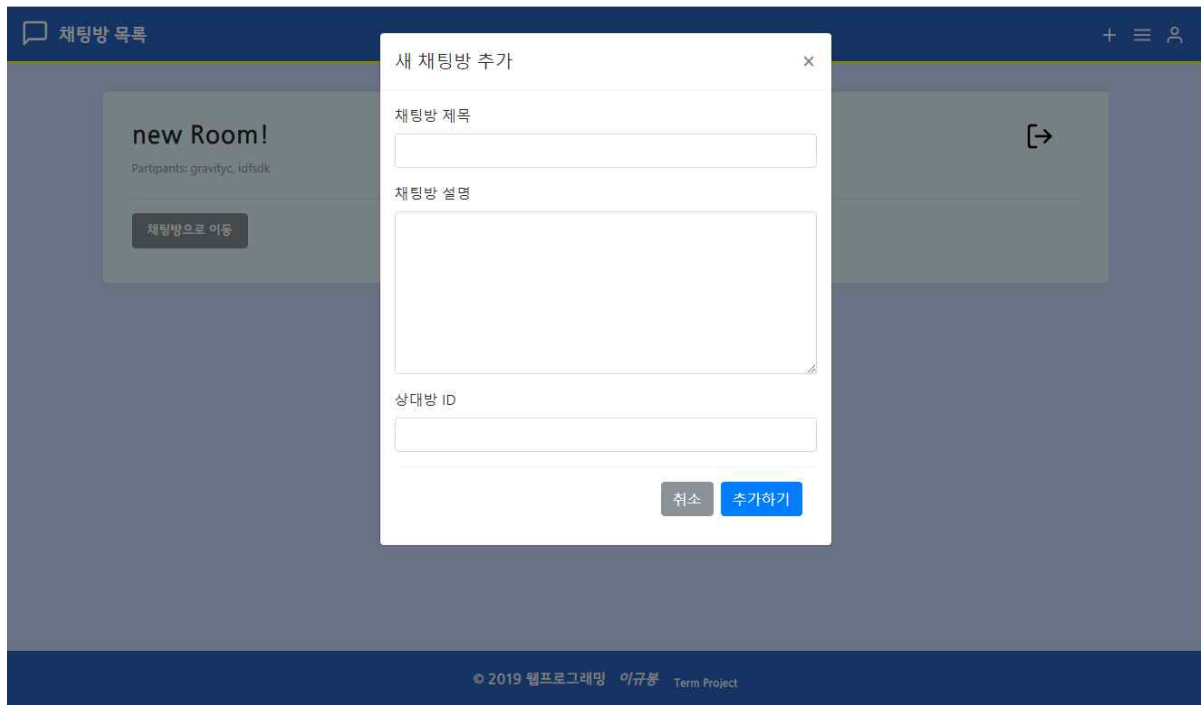
데스크톱에서의 채팅방 선택 화면



스마트폰에서의 채팅방 선택 화면



채팅방 Modal Box 추가 화면



스마트폰에서의 채팅방 Modal Box 추가 화면



채팅방 화면:



위 화면은 gravitytc란 이름의 ID를 가진 유저의 화면이며, 'new Room!' 이란 해당 채팅방의 제목입니다. 자신이 보낸 메시지는 옅은 하늘색으로, 자신 이외의 사람이 보낸 메시지는 짙은 하늘색으로 나타나도록 구별했습니다. DB에 프로필 사진이 등록되어 있지 않은 유저의 프로필 사진은 디폴트 이미지로 지정되도록 했습니다.



위 화면은 같은 유저의 스마트폰에서의 화면입니다.

채팅방 나가기 :

채팅방 나가기 버튼을 누르면 채팅방 나가기 모달 박스가 나옵니다. ‘네’ 버튼을 클릭하면 usersinchattingroom 테이블에서 해당 유저의 정보를 지웁니다. 만약 해당 채팅방이 갖고 있는 유저 레코드가 없다면 해당 채팅방과 채팅방에 해당하는 테이블을 삭제합니다.

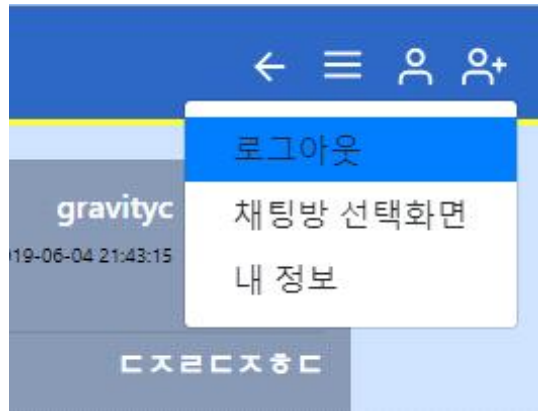


프로필 이미지 모달 박스 :

프로필 이미지 모달 박스는 해당 유저의 ID와 프로필 이미지를 표시해줍니다.



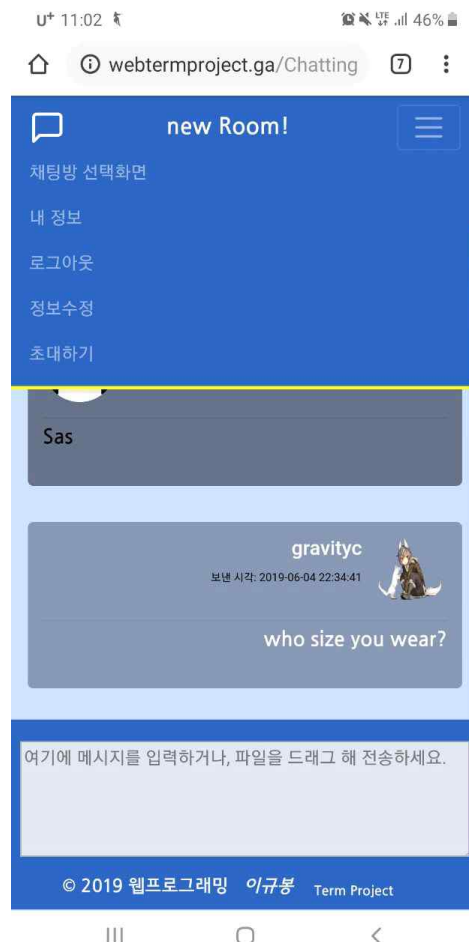
(데스크톱) 드롭 박스 :



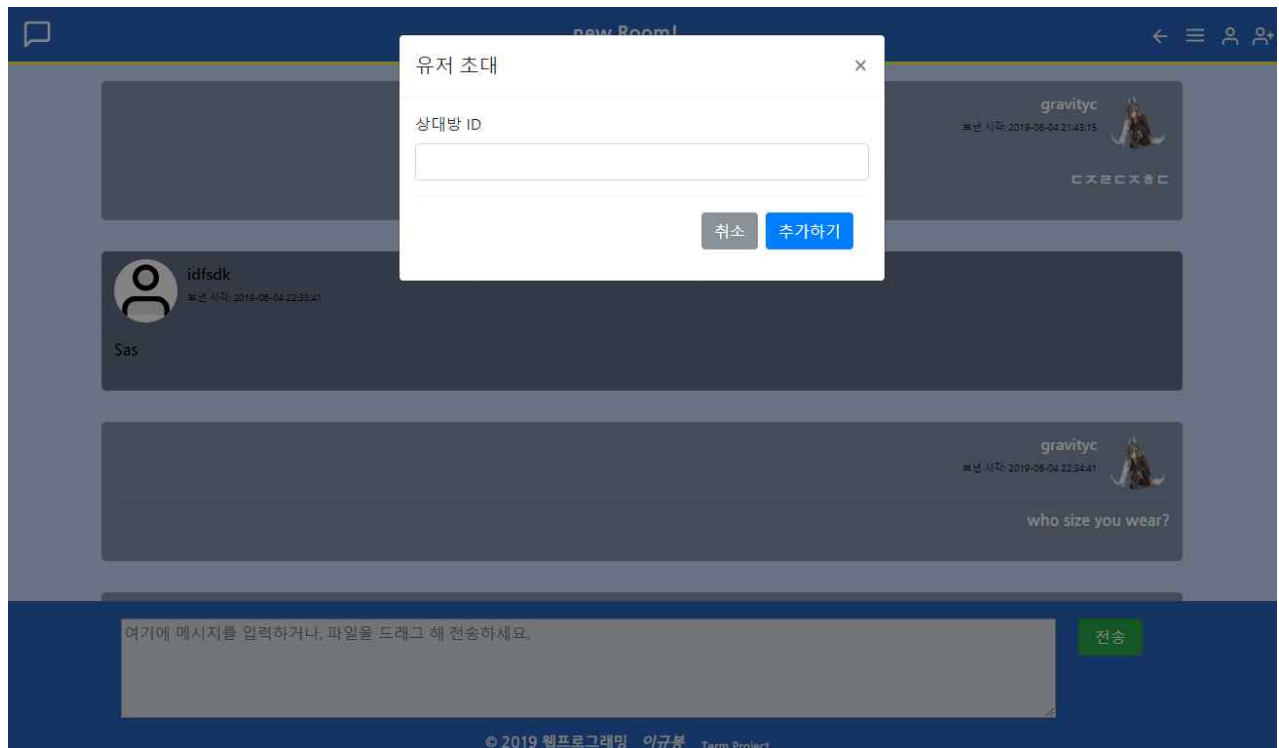
모든 행동을 아이콘으로 만들어 대응시키기 보단, 한 개의 햄버거 버튼을 만들어 드롭 박스 형태를 만드는 것으로 구현했습니다.

(스마트폰) 드롭 리스트 :

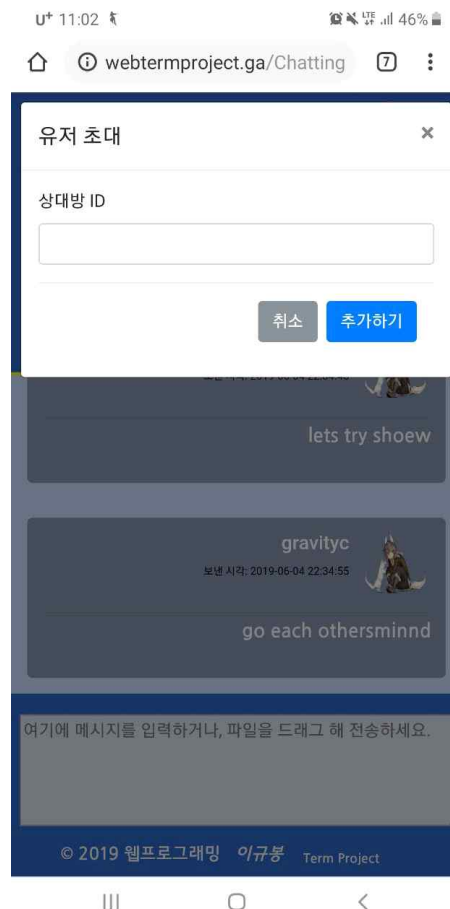
스마트폰과 같은 비교적 작은 화면에선 아이콘들이 아래 줄로 내려오는게 좋아 보이지 않았기 때문에, 하나의 햄버거 버튼을 만들고 햄버거 버튼이 클릭되면 아래로 드롭 리스트가 나오도록 구현했습니다. 스마트폰에선 해당 리스트를 클릭함으로써, 데스크톱에서 아이콘을 눌러 실행하던 일들을 할 수 있습니다.



유저 초대 :



유저 초대 모달박스에서 DB에 존재하는 유저 ID (초대할 수 있는 유저 ID)를 적어 제출하면, usersinchattingroom 테이블에 해당 유저 ID 레코드를 추가해 이 유저가 해당 Room에 들어갈 수 있게 만듭니다.



- 검색 키워드 : ~

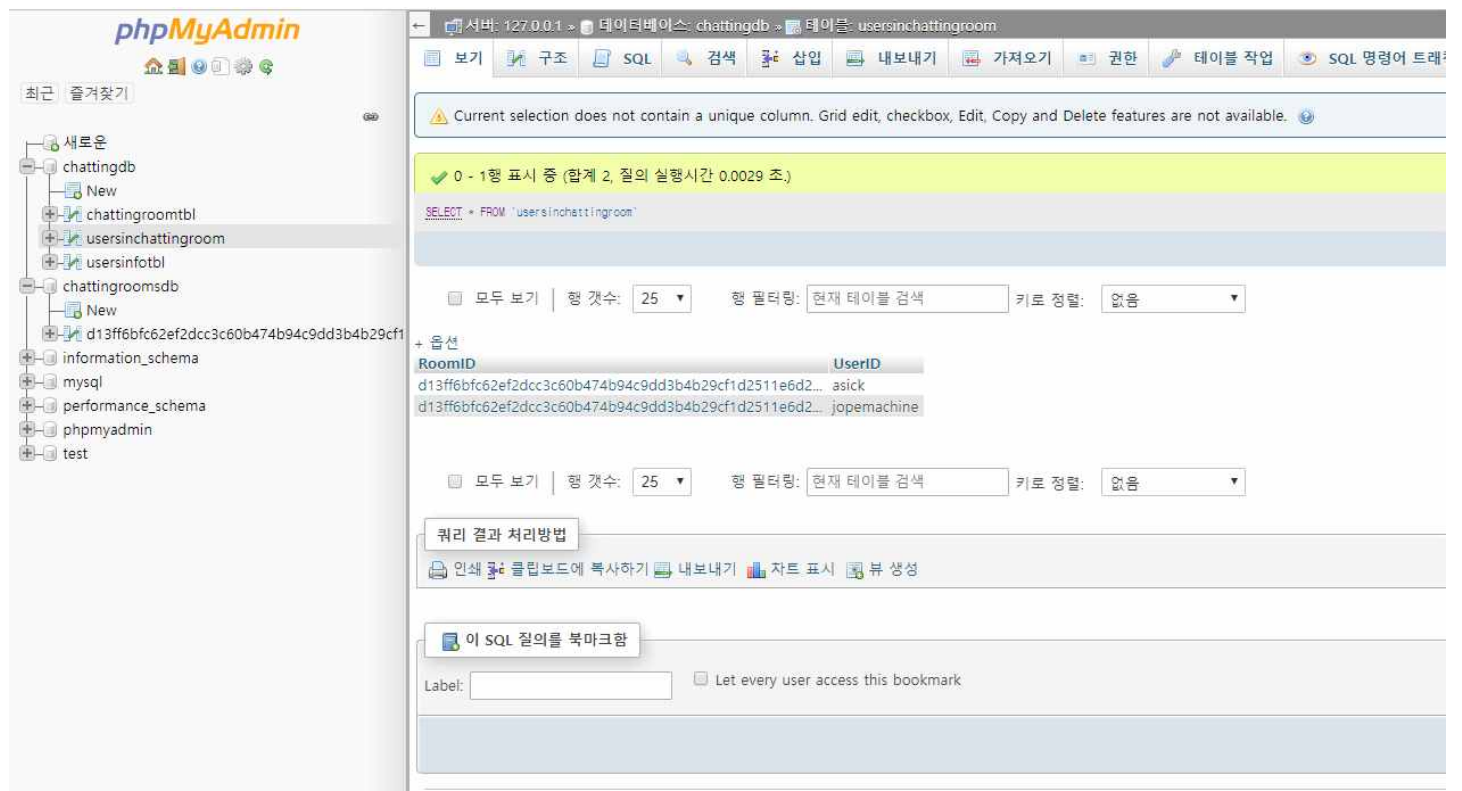
```
<!-- meta 데이터 정의 -->
<meta charset="utf-8">
<meta name="description" content="Chatting web page">
<meta name="keywords" content="Web Programming Term Project, Chatting, chatting Program">
<meta name="author" content="Gyu Bong Lee">
<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1, user-scalable=no" />
```

각각의 페이지들은 위와 같은 keywords, description 등의 meta 데이터를 갖게 함으로써 검색 키워드를 만들어 검색될 수 있게 만들어보려 했습니다.

(2) 사용자 정보 저장

: 제가 사용한 DB는 MySQL입니다. SignUp.html로 폼에 사용자 데이터가 입력되면 SignUpAction.php가 실행되어 데이터들을 MySQL 데이터베이스에 저장합니다. 아래는 사용자의 데이터가 저장되어 있는 예제입니다. 3개의 ID가 저장되어 있습니다. ProfileImageName, Address, PhoneNumber, Gender 등은 필수 데이터가 아닌 데이터이기 때문에, 빈 값 (NULL) 이 저장되어 있을 수 있습니다.

제가 사용한 DB는 아래와 같습니다. (아래 화면은 phpmyadmin이란 웹 프로그램으로, chattingDB 엔 등록되어 있는 모든 유저들의 데이터가 담겨 있습니다.)



The screenshot shows the phpMyAdmin interface. On the left is the database navigation tree with 'chattingdb' selected. The main panel shows the 'usersinchattingroom' table. A message states: 'Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.' Below this, a green bar indicates '0 - 1행 표시 중 (합계 2, 질의 실행시간 0.0029 초)'. The SQL query shown is 'SELECT * FROM `usersinchattingroom`'. The table view shows two rows:

RoomID	UserID
d13ff6bfc62ef2dcc3c60b474b94c9dd3b4b29cf1d2511e6d2...	asick
d13ff6bfc62ef2dcc3c60b474b94c9dd3b4b29cf1d2511e6d2...	jopemachine

Below the table, there are options for '쿼리 결과 처리방법' (Query Result Processing Method) including '인쇄' (Print), '클립보드에 복사하기' (Copy to Clipboard), '내보내기' (Export), '차트 표시' (Show Chart), and '뷰 생성' (Create View). There is also a section for '이 SQL 질의를 북마크함' (Bookmark this SQL query) with a 'Label' input field and a checkbox 'Let every user access this bookmark'.

(3) 프로그램 동작 절차 설명

▶ 회원가입(Sign Up) 절차 설명

- 1) SignUp.html (php 코드가 필요 없으므로 html)에서 form-group에 입력한 데이터들은 submit type의 버튼을 클릭하면 SignUpAction.php로 Post 방식으로 전송되고, SignUpAction.php로 이동합니다.
- 2) Post 방식으로 전송된 데이터들은 슈퍼 전역 변수인, \$_POST에 접근해 가져올 수 있습니다. 이 중 ID를 가져와 chattingdb의 usersinfo table (usersinfotbl)에 접근해 겹치는 값이 있는지 확인합니다.
- 3) 중복되는 ID가 있다면 회원가입은 무효처리 되고, 다시 SignUp.html로 돌아갑니다.
- 4) 중복되는 ID가 없다면 전송된 프로필 사진이 있는지 확인하고, 없는 경우 아무 처리도 하지 않고 넘어갑니다.
- 5) 전송된 프로필 사진이 있다면 프로필 사진을 서버에 업로드하고, mb_stristr로 확장자를 잘라 ID와 붙인 값을 chattingDB의 usersinfo table에 저장해 두고, 나머지 ID, PW, Address, PhoneNumber 등의 정보들도 마찬가지로 같은 테이블에 하나의 레코드로 insert 합니다.
- 6) 로그인 할 수 있도록 SignIn.html으로 이동합니다.

† 회원가입 페이지에 대한 부가 설명

- 1) form-group의 데이터 유효성 검사는, SignUp.js에서 onsubmit을 이용해 했습니다.
- 2) UserID_DuplicateCheck 함수를 SignUp.js에 정의해, ID 중복 검사를 할 수 있는 버튼을 만들었습니다. 회원 가입 도중 다른 페이지로 이동해 유저가 입력해 놓았던 데이터가 사라지면 안 되기 때문에, ajax를 이용해 구현했습니다.
- 3) 프로필 사진 업로드는 canvas 엘리먼트, form-control를 사용한 file type의 input 엘리먼트를 사용해 구현했습니다. 프로필 사진을 파일 Dialog에서 고를 경우, changeProfileImage가 호출되며, FileReader 객체를 생성해 readAsDataURL로 선택한 이미지 파일을 가져옵니다. 그리고 reader의 콜백 함수 (항상 비동기로 실행되는 자바스크립트 환경에서 이미지 파일이 로드된 후에 다음 작업을 진행해야 하기 때문에 onload를 사용해야 합니다)로 가져온 이미지 파일을 canvas에 그려주는 drawImage를 호출해 SignUp.html에 선택된 이미지를 그려줍니다. 파일 업로드 자체는 SignUp.js와 무관하게 php에서 \$_FILES에 접근해 파일을 가져옵니다.

† 프로필 사진 경로에 대한 부가 설명

프로필 사진은 DB에 blob 형태로 저장되어 있지 않습니다. DB엔 프로필 사진의 경로 및 파일의 이름을 저장하며, 프로필 사진 파일은 서버의 profileImages 폴더에 저장되게 해 놓았습니다.

▶ 로그인 절차 설명

- 1) SignIn.html에서 submit type의 버튼을 누르면 SignInAction.php로 이동합니다. 이 때, SignIn.js에서 유효성 검사를 통과하지 못하거나, html 쪽에서 required 필드인 id와 pw가 입력되어 있지 않다면 이동하지 않습니다.
- 2) chattingDB의 usersinfotbl에서 일치하는 레코드가 있는지 검사합니다.
- 3) 일치하는 레코드를 찾으면 PW도 일치하는지 검사합니다.
- 4) 일치하는 경우, setcookie를 이용해 connectedUserID에 1시간 동안, 유저의 로그인 정보를 저장합니다.
- 5) 유저의 프로필 사진의 경로 역시 쿠키로 만들어 함께 저장합니다.
- 6) ChattingRoomSelector.php로 이동합니다.

▶ 로그아웃 절차 설명

로그아웃 기능은 별도의 백엔드 작업이 필요 없기 때문에, Logout.js에서 구현되어 있으며, 제이쿼리 쿠키 플러그인을 사용했습니다. removeCookie로 유저의 로그인 쿠키 및 프로필 사진 정보를 삭제하고, SignIn.html로 이동합니다.

▶ Chatting Room 생성 절차 설명

- 1) 각 유저는 ChattingRoomSelector.php에서 상대 유저 ID를 입력해 채팅방을 생성할 수 있습니다. 상대 ID가 존재하지 않으면 채팅방은 생성되지 않습니다. 채팅방을 만들 땐, 채팅방의 제목이 필요하며, 채팅방의 설명을 추가할 수 있습니다. 채팅방의 제목과 설명은 ChattingRoomSelector.php에 표시 됩니다.
- 2) 채팅방을 생성하면, 채팅방을 생성한 유저의 ID, 유저의 채팅방 인덱스 및 Variant 값 1개를 갖고, sha256 알고리즘으로 Hashing해 RoomID를 만듭니다. (해싱 충돌이 나거나, 같은 RoomID 가 생성되면 Variant 값을 바꿔 다시 해싱합니다.)
- 3) RoomID에 해당하는 테이블을 SQL 문으로 chattingroomsdb에 생성합니다. 이 테이블은 이 채팅 룸에서 오간 메시지를 저장할 테이블이 됩니다.
- 4) chattingroomtbl에 방 레코드를 추가합니다. chattingroomtbl의 레코드 개수는 전체 DB 내의 채팅방 개수와 같습니다.
- 5) usersinchattingroom은, 어떤 유저가 어떤 채팅방들을 갖고 있는지를 나타내는 테이블입니다. 따라서, 채팅방이 하나 생성되면 상대측 ID와 채팅방을 만든 유저의 ID 두 개의 레코드가 usersinchattingroom에 추가 됩니다.

▶ 메시지 전송 절차 설명

- 1) ChattingRoom.php에서 문자 데이터를 입력한 후, 엔터를 누르거나 전송 버튼을 누르면 제 서버의 공유기 IP 주소로 Ajax로 데이터를 전송하는 ChattingRoom.js 내의 함수가 실행 됩니다.
- 2) Ajax로 전송된 데이터는 (파일, 문자 포함) 외부 ip 주소로 전송되고, 외부 ip 주소와 포트 포워딩 되어 있는 사설 IP 주소로 데이터가 전송되게 됩니다.
- 3) 서버의 80 포트에 데이터가 전송되게 되며, 80 포트엔 Apache 가 실행되고 있습니다.
- 4) php에서 3306 포트에 연결합니다. 3306 포트엔 MySQL이 실행되고 있습니다.
- 5) php에서 SQL 구문을 실행해, chattingroomsdb의 RoomID에 해당하는 테이블을 찾습니다.
- 6) 전송된 데이터를 DB에 하나의 레코드 형태로 저장합니다.
- 7) 채팅의 상대 측 컴퓨터에선, 1초에 한 번씩 실행되는 콜백 함수에서 새로 생긴 메시지가 있는지 검사하고 있습니다.
- 8) 메시지가 새로 생기면, chattingroomsdb의 RoomID에 해당하는 테이블에서 요청받은 인덱스보다 큰 Message Index (Auto Increment입니다.) 를 가지는 메시지들을 search 합니다.
- 9) 새로 생긴 메시지는 MessageFetchAction에서 Ajax로 MessageWindow의 static 함수를 거쳐, jumbotron으로 만들어져 상대 측 컴퓨터 (클라이언트) 및 자신의 컴퓨터로 전송됩니다.
- 10) 두 컴퓨터에 각각 새로 전송된 메시지가 표시 됩니다.

▶ 유저 Modal box에 대한 설명

- 1) 쿠키의 프로필 사진 정보를 가져와 img 태그의 src에 적어 주는 형식으로 구현했습니다. img의 src에 경로를 기재하는 식으로 프로필 이미지를 구현하기 위해선, 이미지 파일이 DB가 아닌 서버에 찾을 수 있는 형태로 존재해야 하기 때문에, DB엔 경로만을 저장하고, 서버에 이미지 파일을 업로드 하게 했습니다.

▶ 파일 전송

채팅 전송 박스에 파일을 드래그 해 올려 놓으면 `fileUploadByDrag`가 실행됩니다. 이 함수는 `dataTransfer.files[0].name`으로 파일 이름을 가져 옵니다. (1개 이상의 파일을 드래그 해 실행하면 한 번에 한 파일만 전송할 수 있다는 alert를 띄웠습니다.) 가져온 파일 이름으로 `Response` 객체를 생성해, `arrayBuffer()`로 파일의 바이너리 데이터를 array로 만듭니다. 그리고 `Uint8Array()`를 이용해 바이너리 데이터를 `Unsigned int`로 만들고, `TextDecoder`로 `utf-16`으로 인코딩한 후, `fileUpload` 함수에 이 배열 데이터와 파일 이름 데이터를 넘겨줍니다. `fileUpload` 함수는 `jquery`의 `ajax` 메서드로 `SendMessageActionWithAjax.php`에 넘겨받은 데이터를 전송합니다. 전송된 파일 데이터는 `SendMessageActionWithAjax.php`에서 `Blob` 형태로 메시지 형태로 저장됩니다. 메시지 형태로 레코드를 저장했기 때문에, 다른 메시지들과 같이 `jumbotron` 안에 `a` 태그가 감싸진 형태로 나타나게 되며, 이 `a` 태그를 클릭함으로써, 채팅방 내 모든 유저들은 해당 파일을 다운로드할 수 있게 됩니다. 그러나, 현재 **Multibyte 파일의 인코딩, 디코딩 과정에서 문제가 발생해, MySQL에서 다운로드 받은 파일이 깨져 올바른 형태로 복원되지 않는 버그가 있습니다.**

3. 참고한 코드 및 웹 페이지, 라이브러리

※ 공개된 소스의 이용을 포함하여, 웹에서 제가 참고한 내용은 일단 전부 기재했습니다. 웹, 다른 강의 등에서 공개된 소스를 거의 그대로 가져다 사용한 경우는 빨간색으로 표기하되, 라이브러리를 참고한 경우는 굵게만 표기했습니다. 더 참조한 코드, 라이브러리가 생기면 최종 보고서에 함께 모두 기재하도록 하겠습니다.

1) Bootstrap 4

: 보다 용이한 반응형 웹 페이지 디자인을 위해 사용했습니다. bootstrap.min.css (서버의 부하를 최소화 하기 위한 난독화 파일) 을 사용했으나, css 파일이 어떻게 구성되어 있는지 보기 위해 참고용으로 bootstrap.css를 함께 프로젝트 폴더에 넣어 놓아 참고하였습니다. 일부 코드는 부트스트랩 API를 참고해 작성했습니다. 마찬가지로 이유로 bootstrap.js도 함께 참고했습니다. 또한, 부트스트랩 4의 일부 컴포넌트들은 내부적으로 popper.js, mdb.js를 사용하기 때문에, 제 프로젝트에는 이 파일들도 함께 임포트 되어 있습니다. (CDN 방식으로 사용하지 않을 때 임포트 하지 않으면 에러를 일으킴)

<https://getbootstrap.com/>

2) jQuery

: 자바스크립트 문법을 간결하게 쓰고 싶어 사용했습니다.

<https://jquery.com/>

3) jQuery cookie plugin

: 자바스크립트로 쿠키를 간결하게 다룰 수 있게 해 주는 플러그인입니다.

<https://github.com/carhartl/jquery-cookie>

4) Bootstrap 4 Radio Button 예제

: 보다 예쁜 Radio 버튼을 만들고 싶었기 때문에, 아래 페이지에서 예제 소스를 그대로 가져와 사용했습니다.

<https://getbootstrap.com/docs/4.3/components/buttons/#checkbox-and-radio-buttons>

5) Bootstrap 강의

: 작성한 html 파일 중 부트스트랩에 관련된 부분은 아래 강의에서 배운 것들을 활용한 것이 많습니다. 이 중 특히, **Dropbox, ModalBox**에 관련된 코드는 아래 강의를 들으며 사용했던 프로젝트에서 소스를 가져와 거의 그대로 사용했습니다.

<https://www.udemy.com/bootstrap-4-web-design-tutorial/>

6) freenom

: 웹서버에 무료 도메인을 주기 위해 사용했습니다. 아래 사이트를 알게 된 경로는 생활 코딩 동영상입니다.

<https://www.freenom.com/en/index.html?lang=en>

7) 이것이 MySQL 이다

: PHP에서 DB를 연결하는 코드를 아래 동영상 강의를 들으며 참고해 작성했습니다.

https://www.youtube.com/watch?v=KLZWDOK8kZM&list=PLVsNizTWUw7HhYtI-4GGmlJ5yxNdwNI_X

8) 포트 포워딩

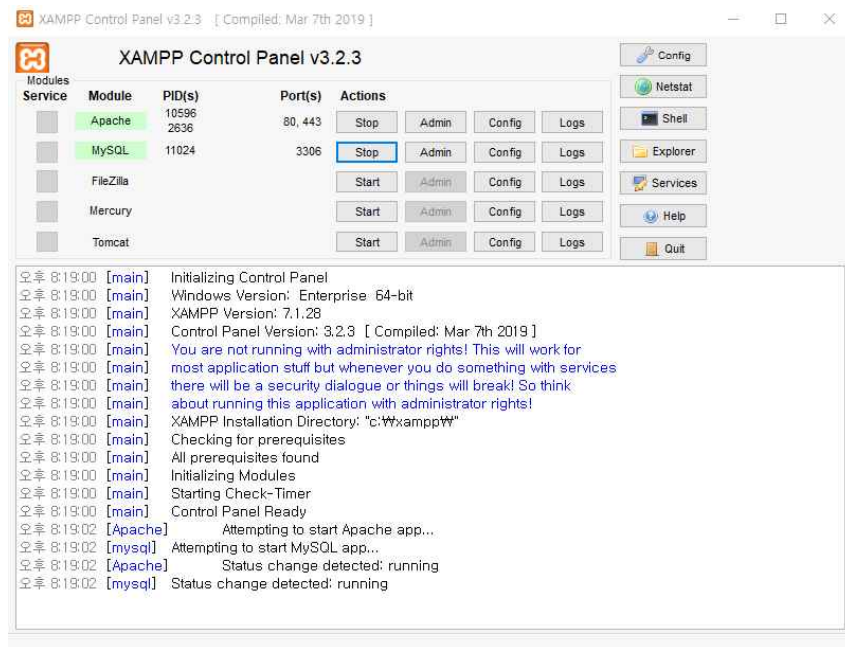
: 집에서 공유기를 사용해 웹서버를 구축하기 위해 사용했습니다. 공유기를 사용하기 때문에 포트 포워딩 과정이 필요합니다. 이에 대한 내용은 생활코딩 사이트를 참고했습니다. 또한, 유동적인 IP를 사용하면 서버 IP가 변경되어 버리기 때문에, 고정 IP로 설정해 사용했습니다.

<https://opentutorials.org/course/3265/20038>

9) XAMPP

: 서버 환경을 용이하게 구축하기 위해 사용한 프로그램입니다. Apache, PHP, MySQL을 동시에 설치해주는 번들형 소프트웨어입니다. 최신 버전의 XAMPP는 MySQL 대신 MariaDB를 사용하지만 사용법은 거의 동일합니다.

<https://www.apachefriends.org/index.html>



10) MySQL

: 회원 데이터를 저장하고 조회하는데 사용한 데이터베이스 입니다

<https://www.mysql.com/>

11) SVG 아이콘 사이트 (feathericons.com)

: 프로젝트에 사용된 svg 파일들은 모두 아래 홈페이지에서 다운로드 받아 사용한 것입니다. 일부 아이콘들은 컬러 값이나, 배경 값, 너비나 높이 등 속성을 변경해 사용했습니다.

<https://feathericons.com/>

12) 폰트

: 프로젝트에 사용된 제주고딕체, 나눔고딕체는 css 파일에서 CDN 방식으로 google apis에서 내려받아 사용한 것입니다. 아래가 그 코드입니다.

@import url(https://fonts.googleapis.com/earlyaccess/jejugothic.css);

@import url(https://fonts.googleapis.com/earlyaccess/nanumgothic.css);

13) 회원가입 페이지 프로필 사진 변경에 대해

: 회원가입 페이지에서 프로필 사진을 클릭할 경우, 이미지 파일이 바뀌는 과정을 아래의 페이지들을 참고하여 작성했습니다. 이 중 특히 FileReader 객체를 생성하는 부분은 복사, 붙여넣기 한 후 변경해 사용했습니다.

https://developer.mozilla.org/ko/docs/Web/HTML/Canvas/Tutorial/Basic_usage

<https://programmingsummaries.tistory.com/367>

https://developer.mozilla.org/ko/docs/Web/HTML/Canvas/Tutorial/Basic_usage

14) Ajax 코드

: Ajax 관련해 코드를 작성할 때 아래의 페이지들을 참고했습니다.

http://tcpschool.com/ajax/ajax_jquery_ajax

<https://marobiana.tistory.com/77>

15) phpmyadmin

: 보다 용이하게 MySQL DB를 조회하거나 관리할 수 있게 해주는 툴 입니다.

<https://www.phpmyadmin.net/>

16) Hamburger Menu 관련해 참고한 API Documents

<https://mdbootstrap.com/docs/jquery/navigation/hamburger-menu/>

<https://getbootstrap.com/docs/4.0/components/navbar/>

17) Modal Box 작성 관련 일부 참고

<https://learnersbucket.com/examples/bootstrap4/custom-confirm-box-with-bootstrap/>

18) 자바스크립트로 파일 다운로드 하기

<https://micropilot.tistory.com/2253>

19) How to convert ArrayBuffer to and from String

<https://developers.google.com/web/updates/2012/06/How-to-convert-ArrayBuffer-to-and-from-String>

20) 자바스크립트 Blob Upload, Response

<https://stackoverflow.com/questions/15341912/how-to-go-from-blob-to-arraybuffer>

21) onDrag 이벤트 관련

<http://blog.naver.com/PostView.nhn?blogId=hyoun1202&logNo=220538438771&parentCategoryNo=&categoryNo=&viewDate=&isShowPopularPosts=false&from=postView>

22) RequireJS

RequireJS는 ReactJS 등에 사용되는 JS 종속성 관리 라이브러리입니다. 파일 다운로드를 구현할 때, require에 콜백 함수를 등록해 사용했습니다.

<https://requirejs.org/>

<https://programmingsummaries.tistory.com/204>

23) FileSaver.js

파일 다운로드를 구현할 때, 사용한 파일 관리 JS 라이브러리입니다.

<https://github.com/eligrey/FileSaver.js>