

2) Node.js 코딩 공유 웹 프로젝트 'Coding-Share'

Node.js 코딩 공유 웹 프로젝트 'Coding-Share'



개발 취지:

개발자들이 프로젝트 진행이나, 코딩 내용을 같이 리뷰하면서 스터디를 진행할 때나 다양한 상황에서 실시간 에디터를 통해 코딩작업을 함께 협업하고 공유해야 하는 상황이 자주 발생했음.

프로젝트 기간 : 2017.09 ~ 2018.01

주요 개발 내용:

실시간 에디터로 코드를 그룹과 함께 협업하고 데이터베이스에 저장하여 필요 시 마다 편집 가능한 것이 주요 기능
웹 소켓을 이용하여 실시간으로 에디터를 공유 및 수정할 수 있게 하였고, 다른 회원에게 코딩방 공유 가능

유일하게 프론트엔드 개발자와 협업한 프로젝트. 서버 1명, 프론트엔드 1명 2명에서 진행

Git Repository : <https://github.com/JisooJang/Node.js-project-CodingShare->

Node.js 코딩 공유 웹 프로젝트 'Coding-Share'

- 데이터베이스 : MongoDB 사용
1. User collection {“_id”: .., “id”: .., “password”:.., “name”:…, “profile_image”:… }
 2. Friend collection {“_id”: .., “id”: .., “friend_id”:..}
 3. Rooms collection {“_id”: .., “participants”: [id array], “contents”: .., “code_language”:…, “room_url”:.., “room_title”:.., “description”:…, “likes”:…, “password”: …, “public”: …, }
 4. Likes collection {“room_url”:.., “user_id”} //회원마다 좋아요 클릭 여부 확인

Node.js 코딩 공유 웹 프로젝트 'Coding-Share'

- 전체 프로젝트 코드 구조

Server.js (서버 실행 및 라우팅, 소켓 연결 및 소켓 이벤트 처리)
`router.route('/myRooms').get(router_function.myRooms);`

Routes/router_function.js

각 라우팅에 들어가는 콜백함수를 모듈화하여 정의

DB와 사용자 기능과 관련된 기능들은 user_function의 모듈화된 함수를 내부적으로 호출

Routes/user_function.js

router_function에 정의된 콜백함수에서 이 user_function 모듈을 불러와

DB와 사용자 관련 기능을 정의한 함수를 호출. (실질적인 DB 쿼리문은 대부분 여기서 실행)

Node.js 코딩 공유 웹 프로젝트 'Coding-Share'

- 전체 프로젝트 코드 구조

- **Views** : 뷰와 관련된 html 정적 페이지를 저장(로그인, 게시판)
- **Js**: 뷰에서 작성한 정적페이지에서 jquery나 자바스크립트 코드를 별도로 분리하여 모듈화한 코드를 저장
- **Uploads** : 사용자가 가입 후 썸네일 이미지 설정 시, 회원이 설정한 이미지들을 저장 후

DB에 각 회원의 profile_image는 해당 이미지경로를 저장

- **Config / passport / facebook.js, naver.js** :

페이스북, 네이버 로그인 기능 사용시 기본적인 클라이언트 확인 정보를 저장하여 모듈화

구현 기능 목록

- 로그인, 회원 가입 기능
- 회원가입 시 폼 입력내용을 DB에 신규로 저장.
- 로그인시 DB에서 입력정보가 존재하는지 체크후 인증 성공 시 서버의 세션에 회원정보(아이디, 닉네임, 프로필이미지) 저장

```
var users = database.collection('users2');    //db에서 users2 이름의 컬렉션 정보를 가져옴

// 아이디와 비밀번호를 사용해 검색
users.find({"id" : id, "password" : password}).toArray(function(err, docs) {
  if(err) {
    callback(err, null); // 콜백함수에 err객체와 null값을 보냄
    return;
  }

  if(docs.length > 0) { // 검색 결과 값이 존재하면
    console.log('아이디 [%s], 비밀번호 [%s]가 일치하는 사용자 찾음.', id, password);
    callback(null, docs); // 콜백함수에 db 검색결과인 docs배열을 전달
  } else { // 검색 결과 값이 존재하지 않으면
    console.log('일치하는 사용자 찾지 못함.');
    callback(null, null);
  }
});
```


회원 검색, 친구 추가, 친구 목록 기능

- 회원 닉네임이나 이메일을 선택하여 검색 후, DB에서 해당 검색 조건에 맞는 데이터 체크. 데이터가 존재하면 화면에 검색 회원정보를 출력
- 각 회원정보의 친구 추가 버튼 클릭 시, 친구 DB에서 세션에 로그인 된 아이디의 친구 목록에 해당 회원이 있는지 체크 후, 없으면 신규로 친구 추가.
- 메인 및 마이페이지에서 DB에 등록된 친구 목록을 확인 가능. (친구 id, 닉네임, 프로필 이미지)

```
var users = database.collection('users2'); //db에서 users2 이름의 컬렉션 정보를
var find_json = {};
find_json[search_option + ""] = value; // json객체에서 key값을 문자열로 표시하기

console.log(find_json);

users.find(find_json).toArray(function(err, docs) {
  if(err) {
    callback(err, null); // 콜백함수에 err객체와 null값을 보냄
    return;
  }
  if(docs) {
    console.log(search_option + " : " + value + " ===== 일치하는 사용자 찾음 ");
    callback(null, docs); // 콜백함수에 db 검색결과인 docs배열을 전달
  } else {
    console.log('일치하는 사용자 찾지 못함. ');
    callback(null, null);
  }
});
```


구현 기능 목록

- 실시간 방 만들기, 실시간 방의 코드 공유

(소켓을 이용하여 실시간으로 키보드 입력시 서버로 소켓 이벤트를 전송.

서버에서는 소켓이벤트를 받을 때마다 다른 방 참여자에게도 입력 데이터를 emit하여 실시간 공유 가능)

```
socket.on("send", function(data) {    // 소켓에 "send" 이벤트 연결
  client_info = {
    contents: data.contents,
    cursor: data.cursor
  };
  console.log('Data : ' + data.contents + 'cursor : ' + data.cursor);
  socket.in(roomName).emit('get', client_info);    // 참여중인 방에 소켓 데이터 전송
});

socket.on('chat', function(data) {    // 코드공유 페이지 내 멤버 채팅 이벤트 처리
  console.log('room Name : ' + roomName);
  console.log('Data : ' + data);

  socket.in(roomName).emit('chat_get', data);
});
```

(서버 소켓 코드)

구현 기능 목록

- 실시간 방 만들기, 실시간 방의 코드 공유

(소켓을 이용하여 실시간으로 키보드 입력 시 서버로 소켓 이벤트를 전송.

서버에서는 소켓이벤트를 받을 때마다 다른 방 참여자에게도 입력 데이터를 emit하여 실시간 공유 가능)

```
socket.on('get',function(data){
  console.log('get data : ' + data);
  $(editor.document.body).html(data.contents);

  if(data.cursor<lastCursor){
    lastCursor++;
  }
  var iframeElement = document.getElementById('editor'); // the
  var contentDoc = iframeElement.contentDocument;
  var range = contentDoc.createRange();
  range.setStart(contentDoc.body.firstChild, lastCursor);
  range.setEnd(contentDoc.body.firstChild, lastCursor);
  var selection = iframeElement.contentWindow.getSelection();
  selection.removeAllRanges();
  selection.addRange(range);

});|
```

(클라이언트 소켓 코드)

구현 기능 목록

방 수정 내용 저장 / 방 신규 저장 기능

(방 저장 시 해당 방이 DB에 존재하면 방 내용만을 수정하고, DB의 참여자에 본인 id만을 추가
만약 방이 DB에 없으면 DB에 방 레코드 전체를 새롭게 추가)

```
var room = database.collection('rooms');
room.update({"room_url": room_url}, { $addToSet : { "participants" : user_id }
  if(err) { throw err; }
  if(docs) {
    console.log('방 참여자 추가 성공');
  } else {
    console.log('방 참여자 추가 실패');
  }
});
room.update({"room_url": room_url}, { $set : { "contents" : contents }}, funct
  if(err) { throw err; }
  if(docs) {
    console.log('방 내용 수정 성공');
    callback(null, docs);
  } else {
    console.log('방 내용 수정 실패');
    callback(null, null);
  }
}
```


구현 기능 목록

- 방 상세 정보 수정 기능

먼저 DB에 해당 방이 있는지 체크 후, DB에 입력된 내용

(방 제목, 방 설명, 방 비밀번호, 방 입력 내용, 코드 언어, 공유 여부)을 update

```
if(req.session.user) {  
  if(database) {  
    user.modify_room_submit(database, room_id, room_title, description, password, code_lang  
    if(err) { throw err; }  
    if(docs) {  
      res.send({'alert_message' : 'success'});  
    } else {  
      res.send({'alert_message' : '데이터 수정 오류입니다. 잠시 후 다시 시도해주세요.'});  
    }  
  }  
});  
}
```


구현 기능 목록

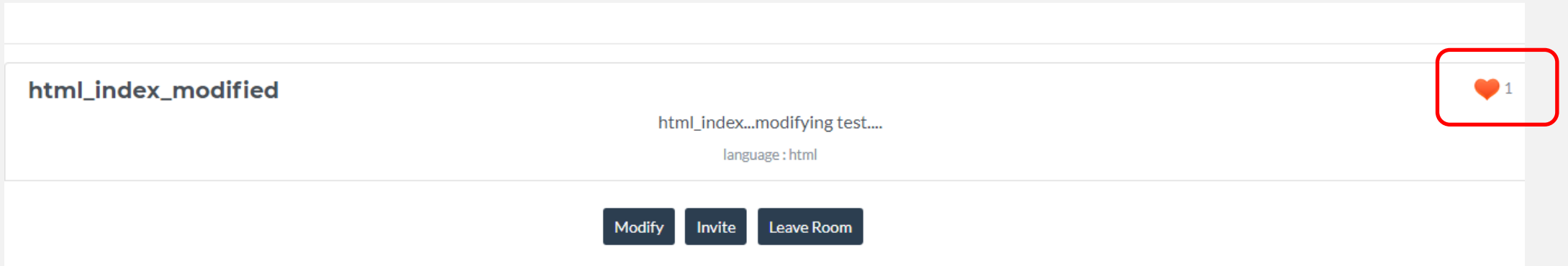
- 다른 회원에게 방 초대 기능

회원 이메일을 입력 받아서, 해당 이메일로 가입된 사용자가 있는지 DB에서 먼저 체크 후,
존재하는 회원이면 고유의 방 URL 링크를 입력 받은 메일로 전송하여 초대

```
if(result_code == 1) {
    user.shareRoom(database, room_url.substring(32, room_url.length), function(err, docs) {
        if(err) { throw err; }
        if(docs.length > 0) {
            console.log('방 가져오기 성공');
            console.log(JSON.stringify(docs));

            html += '방 이름 : ' + docs[0].room_title;
            html += '<br>방 설명 : ' + docs[0].description + '<br>';
            html += '<a href="' + room_url + '>방 입장하기</a>';
            console.log(html);
            user.sendEmail(req.session.user.id, receiver, 'Coding-chat 실시간 코딩방 초대 메일');
        }
    });
    res.send({'alert_message' : 'success'});
} else {
    res.send({'alert_message' : '작성 아이디는 회원목록에 존재하지 않습니다.'});
}
```

구현 기능 목록



- 방 좋아요 기능

- 1) User_id, room_id 정보를 서버에 보내서, **우선 Likes DB에 {user_id, room_id} 레코드가 존재하는지 체크**
- 2) **레코드가 존재하면**, 이미 좋아요를 체크한 상태로 간주하여 Rooms DB에서 해당 room_id로 방 좋아요 수를 찾고, 찾은 방 좋아요 수에서 -1을 적용한 뒤 다시 DB update

좋아요를 취소한 상황이므로, Likes DB에서 해당 레코드 {user_id, room_id} 삭제

- 3) **레코드가 존재하지 않으면**, 좋아요를 체크한 기록이 없는 상태로 간주하여 Rooms DB에서 해당 room_id로 방 좋아요 수를 찾고, 찾은 방 좋아요 수에서 +1을 적용한 뒤 다시 DB update

Likes DB에서 해당 레코드 {user_id, room_id} 추가

구현 기능 목록

- 방 좋아요 기능 (Likes DB에 내용이 존재하는지 체크하여 좋아요를 증가할지 감소할지 결정하는 코드)

```
var likeRooms = function(database, user_id, room_id, callback) {
  console.log('likeRooms 호출됨');
  var likes = database.collection('likes');
  var rooms = database.collection('rooms');
  likes.find({'id': user_id, 'room_id': room_id}).toArray(function(err, docs) {
    if(err) { throw err; }
    if(docs.length > 0) {
      console.log('이미 좋아요를 누름');
      modify_likes(database, room_id, 0);

      likes.remove({'id': user_id, 'room_id': room_id}, function(err, docs) {
        console.log('좋아요 레코드삭제 완료');
      });

      callback(null, {'like_result' : 'min'});
    } else {
      console.log('좋아요 반영');
      modify_likes(database, room_id, 1);

      likes.insertMany([{'id': user_id, 'room_id': room_id}], function(err, result) {
        if(err) {
          callback(err, null);
          return;
        }
        if(result.insertCount > 0) {
          console.log('좋아요 체크 레코드 추가됨 : ' + result.insertedCount);
        } else {

```


구현 기능 목록

- 방 좋아요 기능 (DB에서 매개변수 key값에 따라 좋아요 정보를 증가/감소시키는 함수)

```
var modify_likes = function(database, room_id, key) {
  var likes_num = 0;
  console.log('modify_likes 호출됨');
  var rooms = database.collection('rooms');
  rooms.find({'room_url': 'http://127.0.0.1:3500/shareRoom/' + room_id}).toArray(function(err, docs) {
    if(err) { throw err; }
    if(docs) {
      likes_num = parseInt(docs[0].likes);
      console.log('likes_num : ' + likes_num);
    }
  });
  if(key == 1) {
    likes_num = likes_num + 1;
    console.log('likes_num2 : ' + likes_num);
    rooms.update({'room_url': 'http://127.0.0.1:3500/shareRoom/' + room_id}, {$set: {'likes': likes_num}, {}});
  } else {
    likes_num = likes_num - 1;
    console.log('likes_num2 : ' + likes_num);
    rooms.update({'room_url': 'http://127.0.0.1:3500/shareRoom/' + room_id}, {$set: {'likes': likes_num}, {}});
  }
}
```


구현 기능 목록

회원가입 및 비밀번호 찾기 시 확인 메일 전송 기능

- node-mailer모듈을 사용하여 메일 전송. **회원 가입 시 사용자 가입정보를 메일로 발송**
- 비밀번호 찾기 시 입력 받은 이메일로 본인 인증 메일을 1차로 발송 후, 인증 메일에서 수락버튼을 누르면 2차로 임시 비밀번호를 재전송하고 해당 시점에서 회원 DB의 비밀번호를 임시비밀번호로 update

```
if(req.session.user) {
  console.log('이미 로그인되어 있습니다. ');
  res.send({"alert_message" : "이미 로그인되어 있습니다."});
} else {
  var email = req.params.id;
  console.log('findPassword' + email);
  var subject = '코딩쉐어 비밀번호 찾기 안내 메일입니다.';
  var pass_temp = shortid.generate() + shortid.generate();
  var html_content = '코드쉐어에서 회원님의 이메일 계정으로 임시 비밀번호를 보내드립니다.<br>

  if(database) {
    var user2 = database.collection('users2');
    user2.update({'id' : email}, {$set : {'password' : pass_temp}});
  }

  user.sendEmail('inspirebj@gmail.com', email, subject, html_content);
  res.send("임시 비밀번호를 해당 메일로 전송하였습니다. 임시비밀번호를 이용하여 로그인후 비밀번호");
}
```


구현 기능 목록

네이버로 로그인 인증 기능

Passport, passport-naver 모듈을 사용하여 사용자가 네이버의 아이디와 비밀번호를 입력 후 인증 성공시, 네이버의 기본 회원 정보(이메일, 닉네임)를 받아와 서버의 사용자 세션에 저장 -> 마이페이지에서 확인 가능

```
var naver_callback = function(app, passport) {  
  return new NaverStrategy({  
    clientID : config.naver.clientId,  
    clientSecret : config.naver.clientSecret,  
    callbackURL : config.naver.callbackURL  
  }, function(accessToken, refreshToken, profile, done) {  
    console.log('passport의 naver 호출됨');  
    console.dir('profile:' + profile);  
  
    var _profile = profile._json;  
  
    var user = {  
      'auth_type' : 'naver',  
      'auth_id' : _profile.id,  
      'auth_name' : _profile.nickname,  
      'auth_email' : _profile.email  
    };  
  
    session_user = {  
      id: user.auth_email,
```


구현 기능 목록

네이버로 로그인 인증 기능

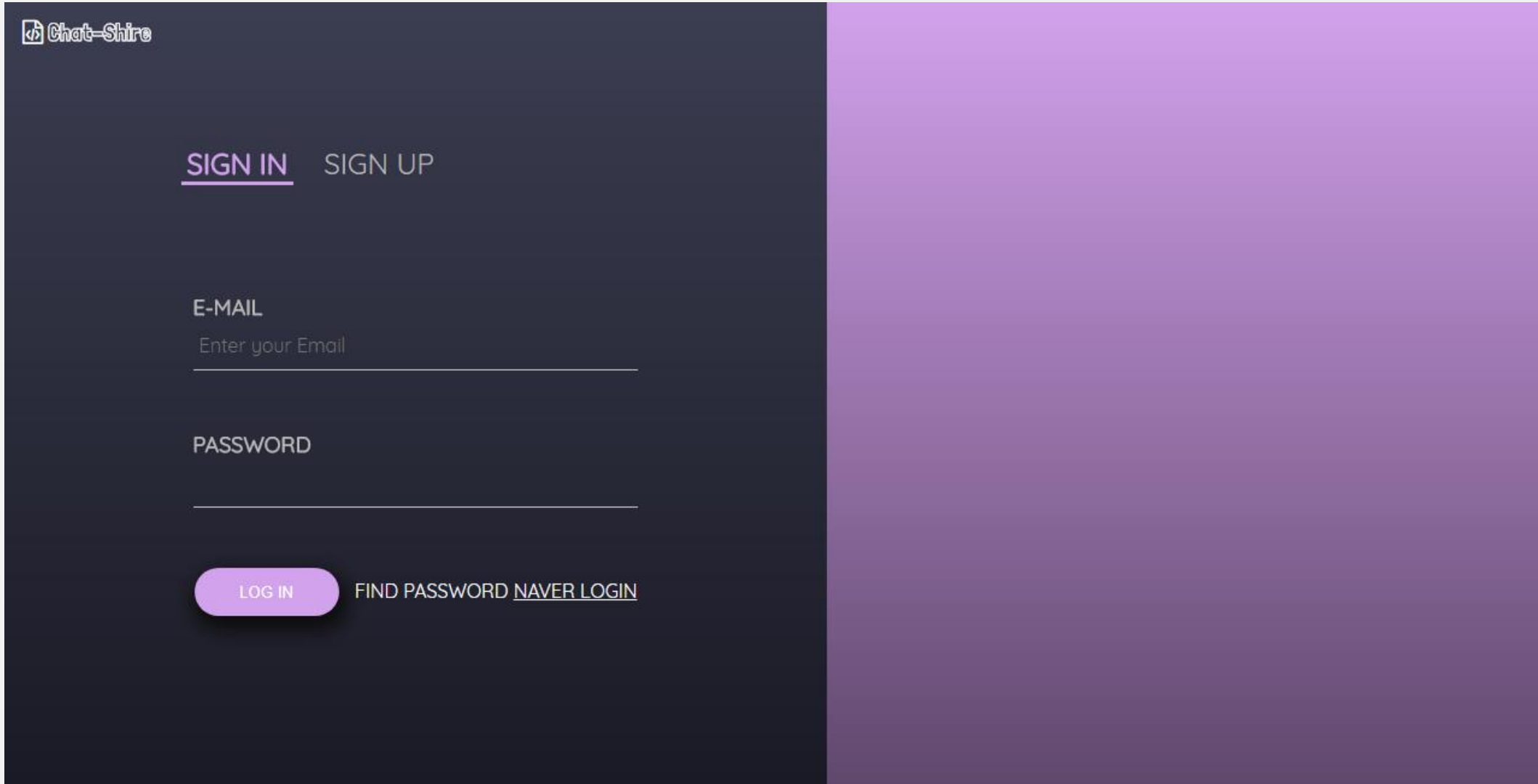
Passport, passport-naver 모듈을 사용하여 사용자가 네이버의 아이디와 비밀번호를 입력 후 인증 성공시, 네이버의 기본 회원 정보(이메일, 닉네임)를 받아와 서버의 사용자 세션에 저장 -> 마이페이지에서 확인 가능

```
session_user = {  
  id: user.auth_email,  
  name: user.auth_name,  
  profile_image: 'http://127.0.0.1:3500/public/image/default_profile.png',  
  authorized: true  
}  
  
done(null, session_user);
```

(네이버 인증 성공 후 네이버 회원 정보를 세션 정보에 저장처리)

Node.js 코딩 공유 웹 프로젝트 'Coding-Share

프로젝트 시연 화면



Chat-Shire

SIGN IN SIGN UP

E-MAIL
Enter your Email

PASSWORD

LOG IN FIND PASSWORD [NAVER LOGIN](#)

최초 화면 - 로그인 페이지
로그인 폼 및 비밀번호 찾기, 네이버 로그인 기능

Chat-Shire

SIGN IN SIGN UP

E-MAIL
Enter your Email

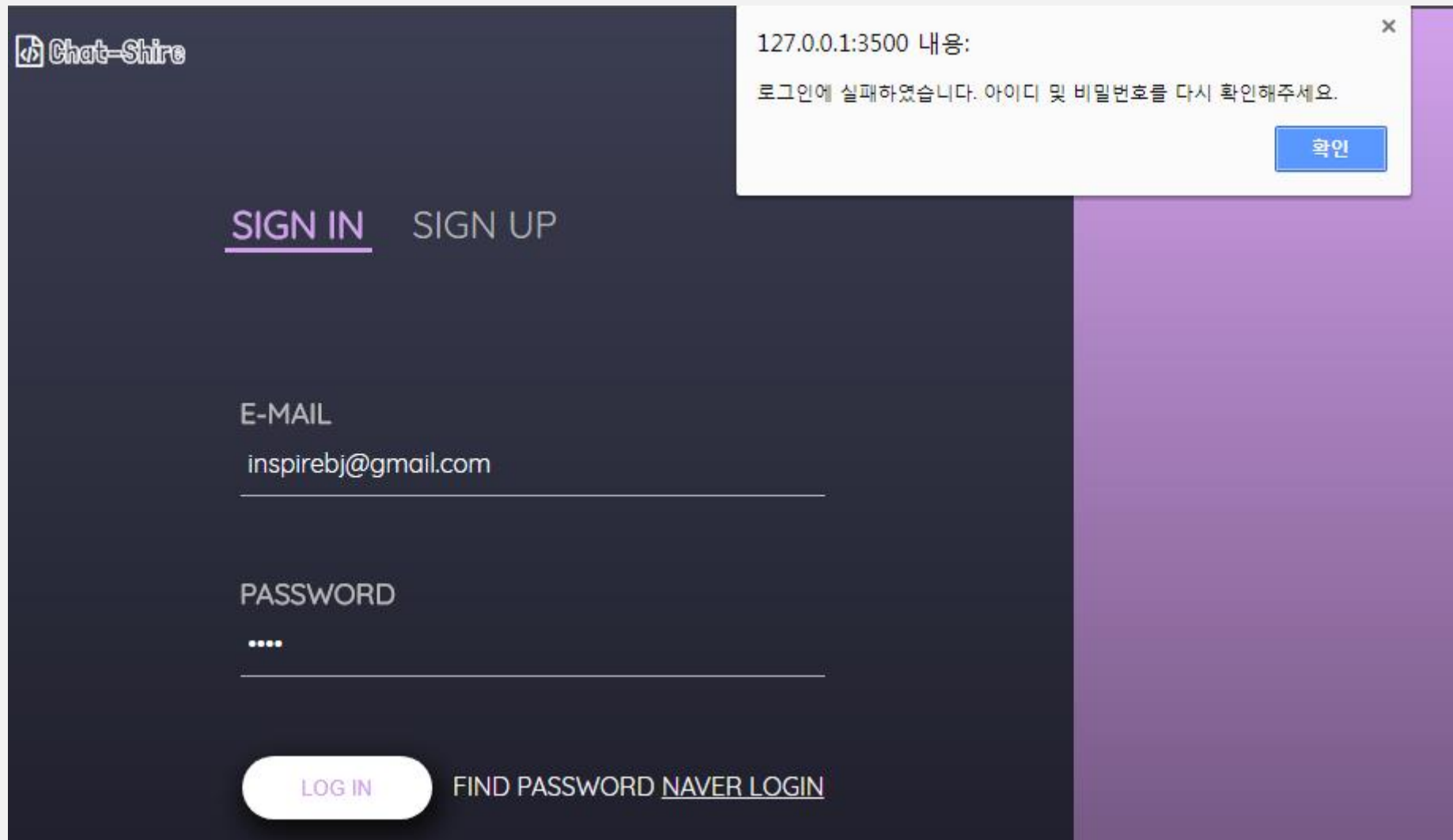
PASSWORD

CONFIRM

NICKNAME
Enter your Nickname

SIGN UP

최초 화면 - 회원가입 페이지, 회원가입 폼 제공



메인 로그인 화면에서 로그인 실패시, alert창으로 실패 여부를 알려준다.



The image shows a Naver login interface. At the top is the Naver logo in green. Below it is a message in Korean: "네이버 로그인으로 **nodejs-project** 서비스를 이용하실 수 있습니다." (You can use the **nodejs-project** service by logging in with Naver). There are two input fields: the first is labeled "아이디" (ID) and the second is labeled "비밀번호" (Password). Below these fields is a large green button with the text "로그인" (Login). At the bottom, there are three items: a checked checkbox labeled "로그인 상태 유지" (Keep login state), the text "IP보안 ON" (IP Security ON), and a link "일회용 로그인 ?" (One-time login ?).

NAVER

네이버 로그인으로 **nodejs-project** 서비스를 이용하실 수 있습니다.

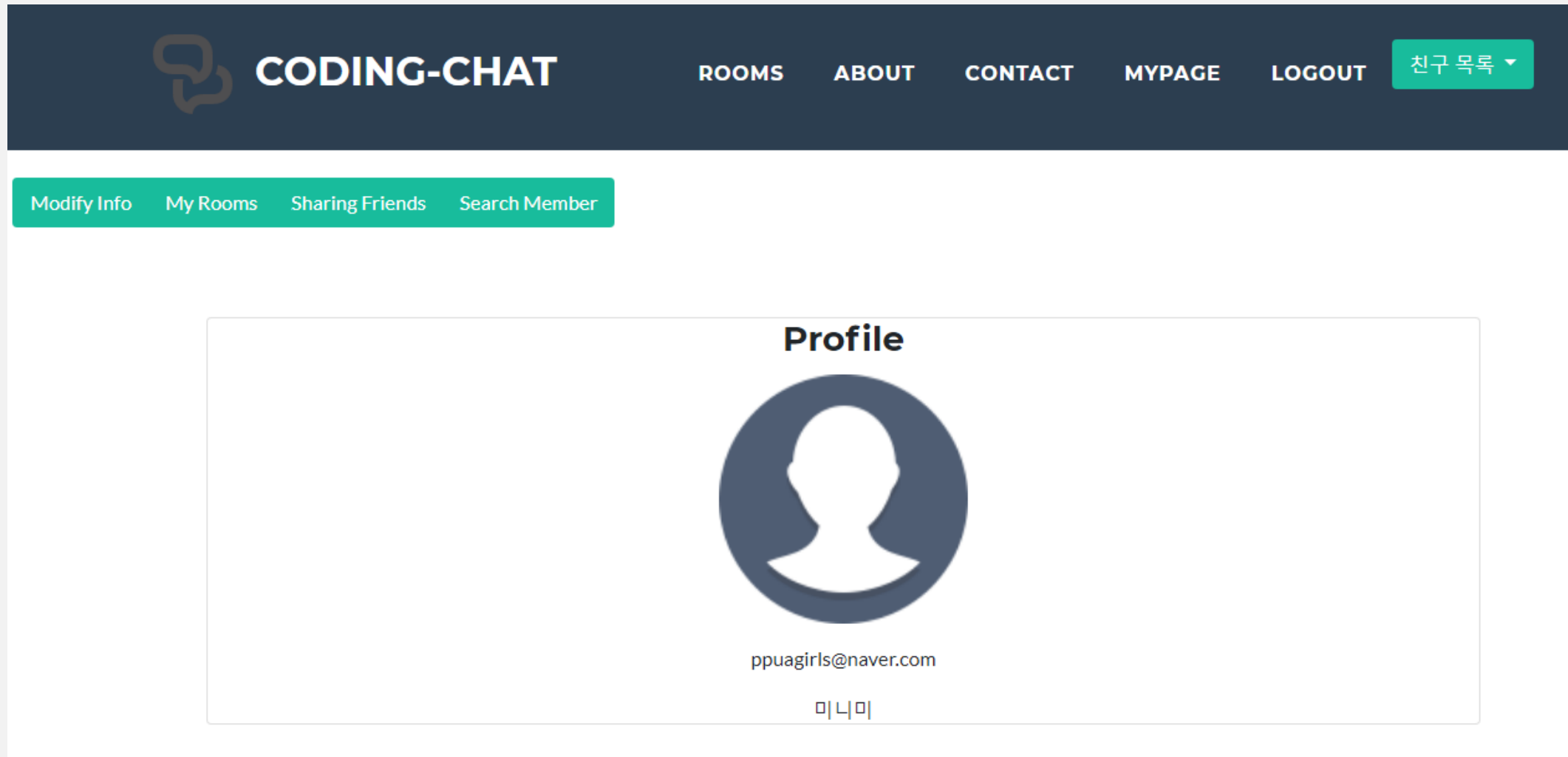
아이디

비밀번호

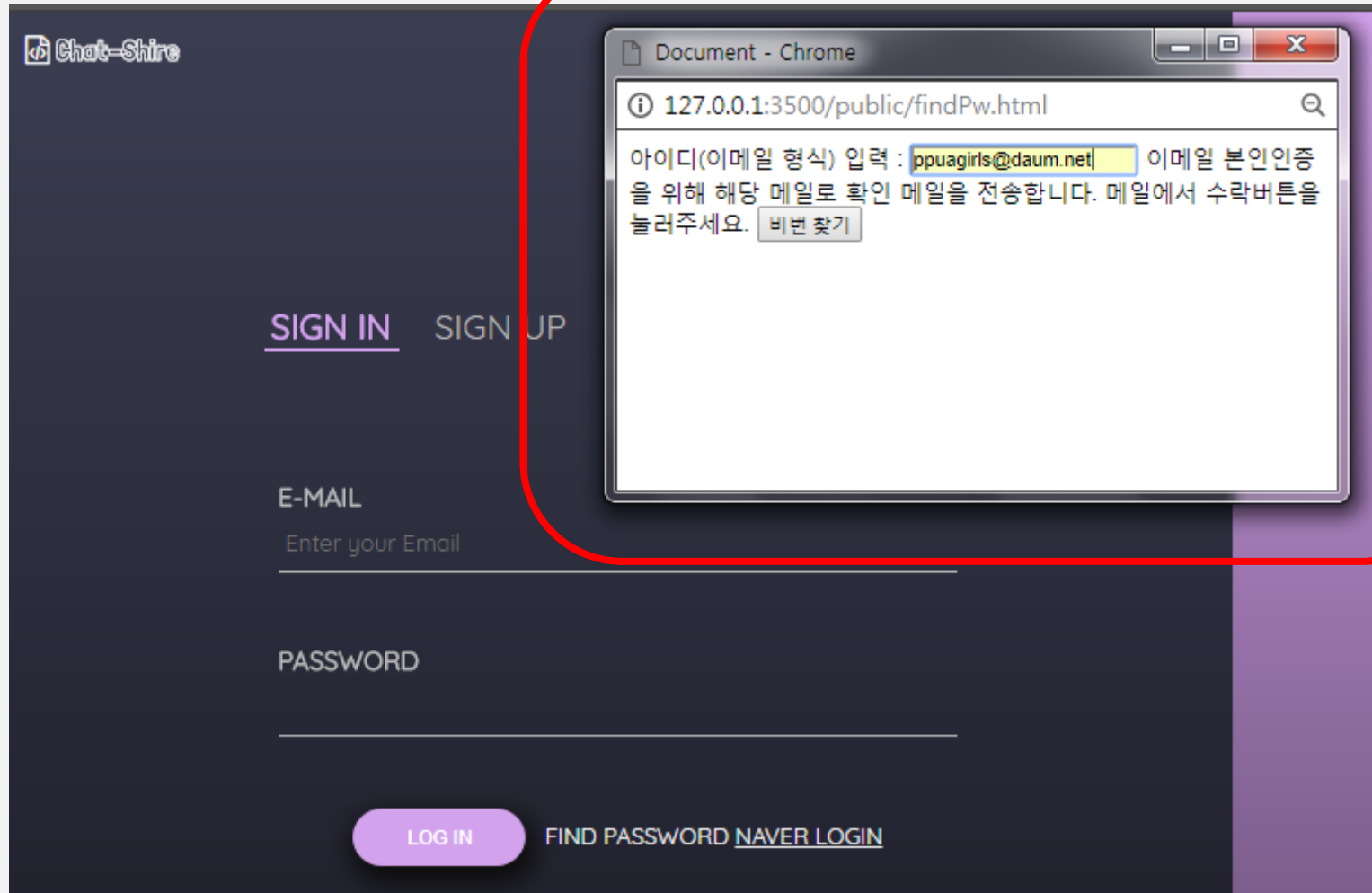
로그인

☒ 로그인 상태 유지 IP보안 **ON** | [일회용 로그인 ?](#)

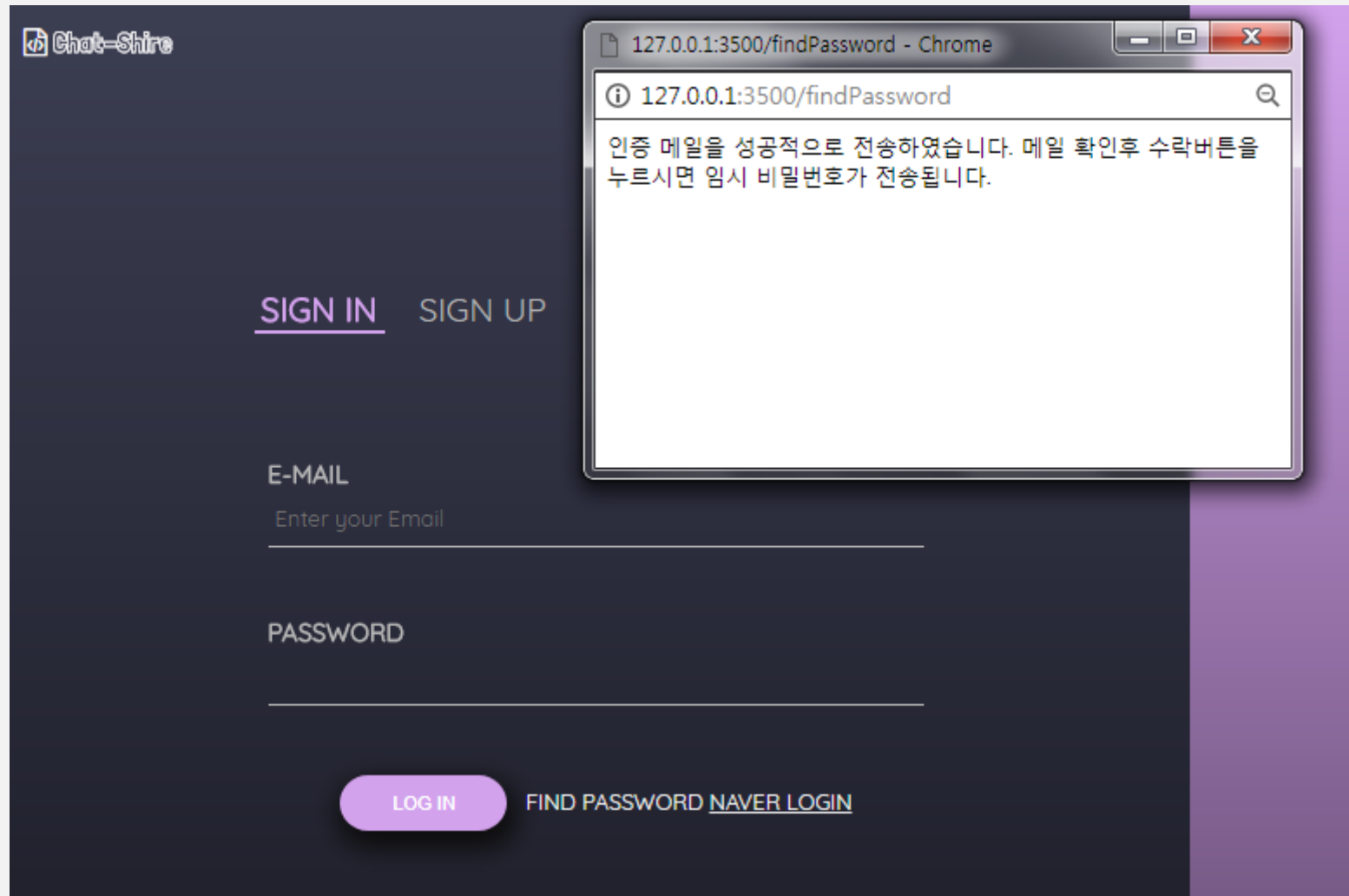
네이버로 로그인 버튼 클릭시 - 네이버 로그인 폼을 새 창으로 보여줌



네이버로 로그인 성공후, 마이페이지 화면
네이버에 저장된 기본 회원정보를 서버의 세션에 저장하여 활용



메인 로그인 화면에서 비밀번호 찾기 버튼 클릭
새 창을 띄워서 검색할 회원의 아이디를 입력하도록 하고,
서버에서 회원 DB에 해당 회원이 존재하는지 먼저 체크한다.



메인 로그인 화면에서 비밀번호 찾기 버튼 클릭
서버에서 회원 DB에 해당 회원이 존재하는지 먼저 체크한 후, 회원이 존재하면
입력한 이메일로 1차 인증메일을 전송한다.



비밀번호 찾기 1차 인증메일 수신 화면

먼저 비밀번호 찾기 동작을 한 사람이 해당 메일의 계정인지 확인하기 위해,
1차 인증메일을 보내어 메일 내에서 수락버튼을 클릭해야지만 비밀번호를 변경시키도록 한다.



수락 버튼 클릭 후(1번째 이미지), 비밀번호 찾기 2차 인증메일 수신 화면 (2번째 이미지)
 2차 인증메일이 전송된 시점에서, 임시비밀번호를 랜덤값으로 생성하여 전송해주고,
 임시비밀번호를 회원 DB에 반영하여 저장한다.

Chat-Shine

[SIGN IN](#) [SIGN UP](#)

E-MAIL
ppuagirls@daum.net

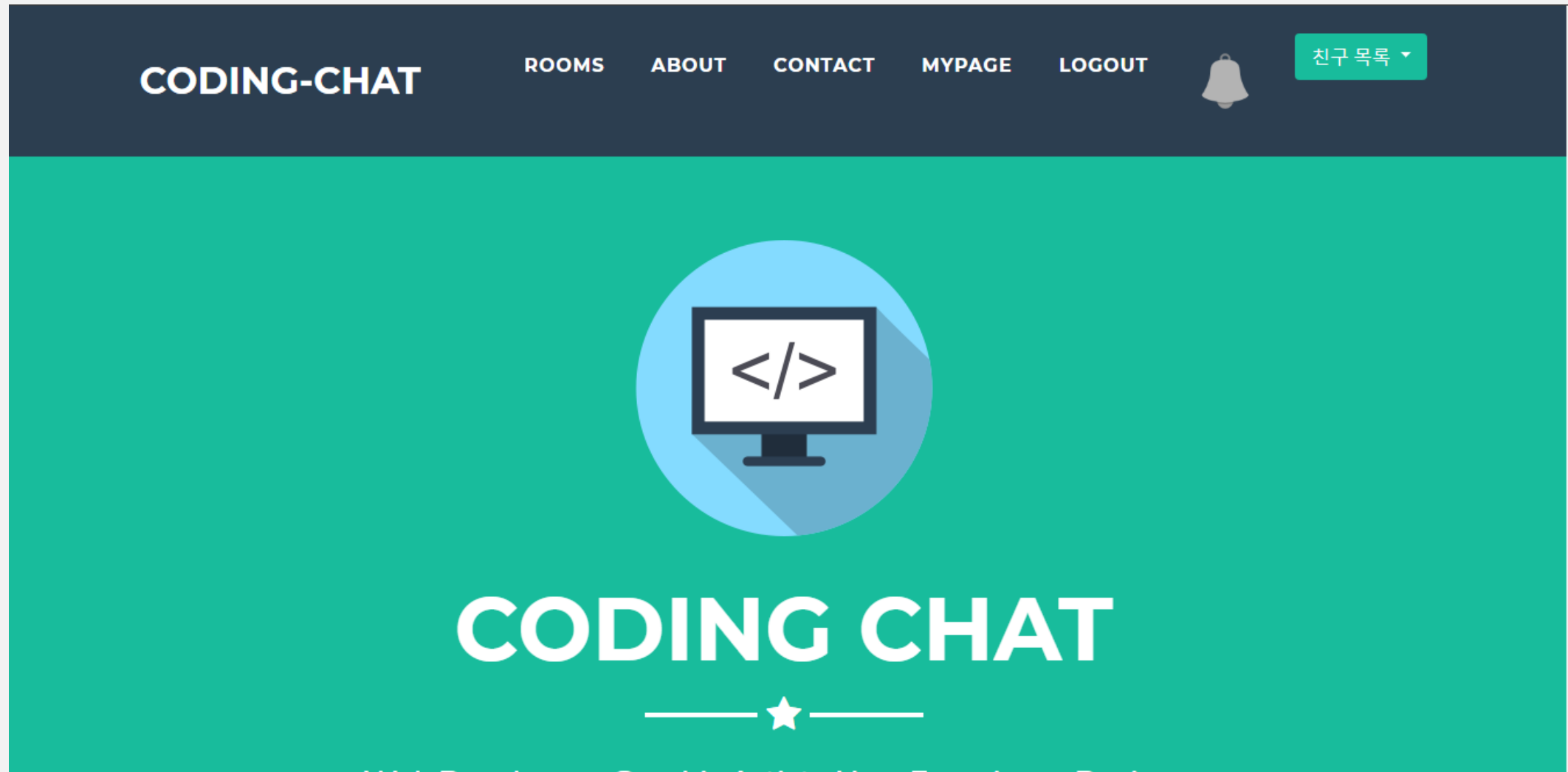
PASSWORD
.....

[LOG IN](#) [FIND PASSWORD](#) [NAVER LOGIN](#)

```
{  
  "_id" : ObjectId("5a3761b41985d71aa4445c26"),  
  "id" : "ppuagirls@daum.net",  
  "password" : "1234",  
  "name" : "Jisoo-Jang",  
  "profile_image" : "http://127.0.0.1:3500/public/image/default_profile.png"  
}
```

```
{  
  "_id" : ObjectId("5a3761b41985d71aa4445c26"),  
  "id" : "ppuagirls@daum.net",  
  "password" : "BJ9H3sg4zHJx5B3jgUf",  
  "name" : "Jisoo-Jang",  
  "profile_image" : "http://127.0.0.1:3500/public/image/default_profile.png"  
}
```

비밀번호 찾기를 통해 임시 비밀번호로 변경이 완료되면,
변경된 임시비밀번호로 로그인 가능하다.
DB에도 변경된 비밀번호가 저장된 것을 확인할 수 있다.



CODING-CHAT

ROOMS

ABOUT

CONTACT

MYPAGE

LOGOUT

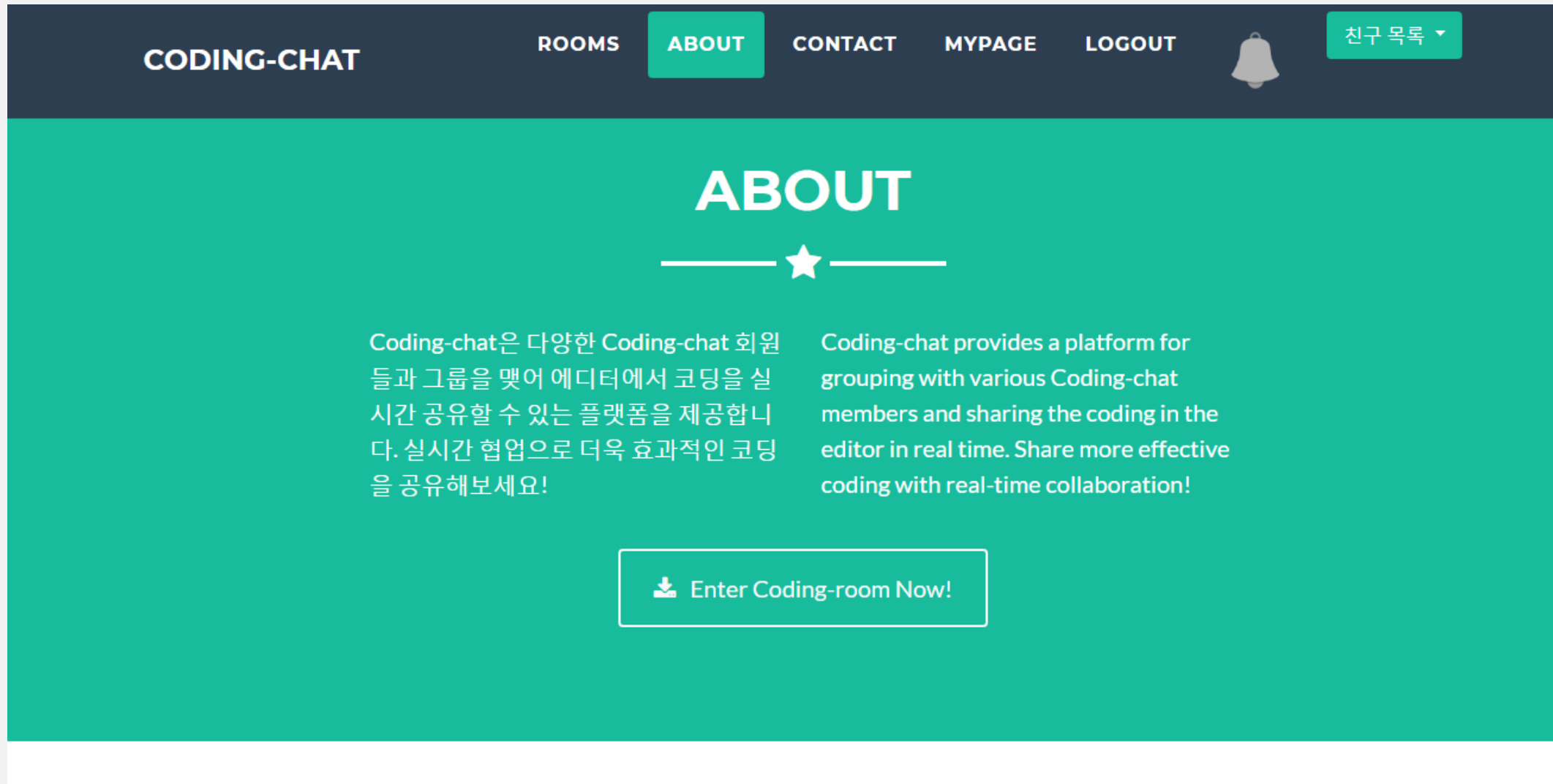


친구 목록 ▾

POPULAR CODING-CHAT ROOMS



메인페이지 2



메인페이지 3 - 홈페이지 정보 설명, 코딩 방 참여 버튼

CODING-CHAT


ROOMS

ABOUT

CONTACT

MYPAGE

LOGOUT



친구 목록 ▾

CONTACT TO CODING-CHAT

————★————

Name

Email Address

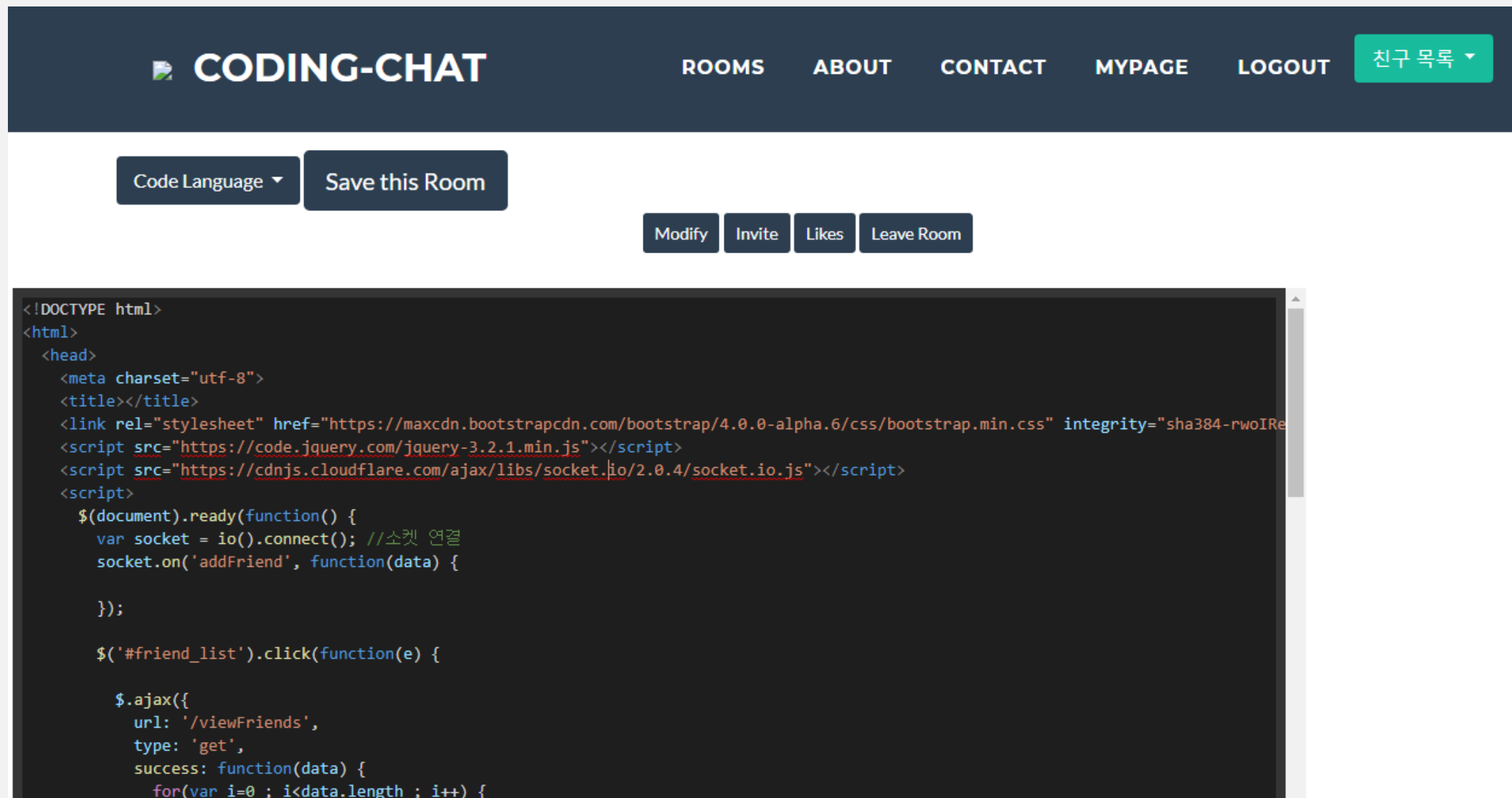
Phone Number

Message

Send

메인페이지 4

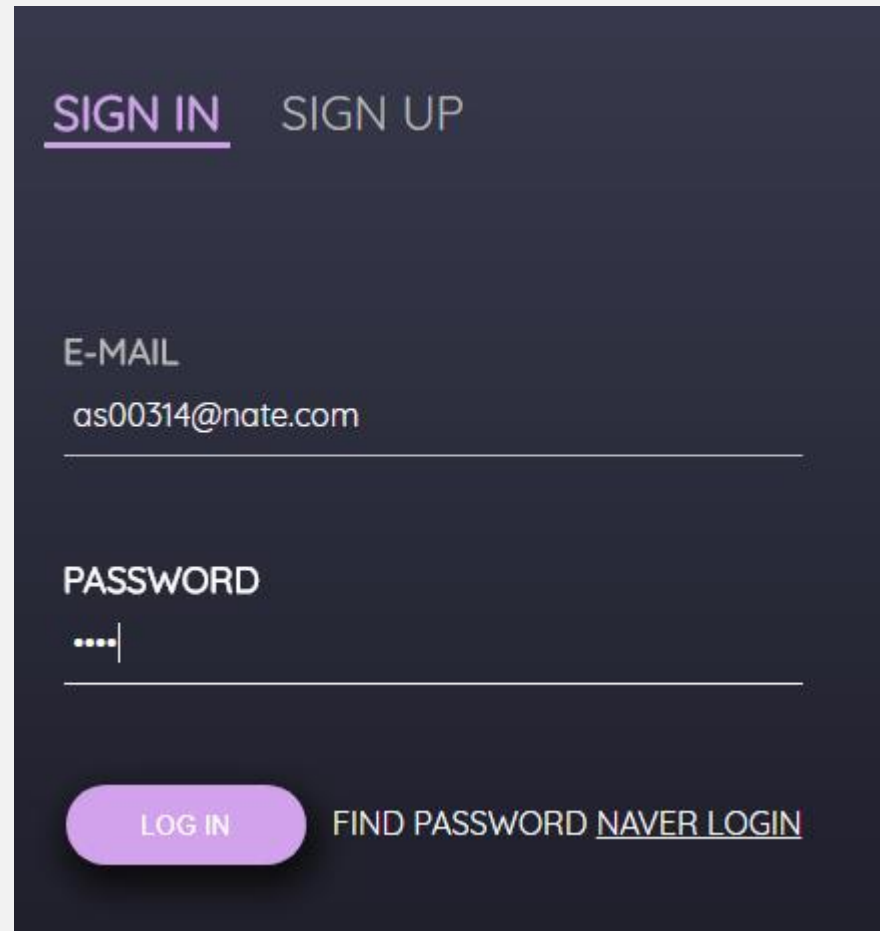
웹 관리자에게 CONTACT. 폼 입력후 전송시 관리자 메일로 해당 내용이 전송됨



메인에서 신규 코딩방 생성 버튼을 클릭

URL - <http://serverURL/shareRoom/H1Akn2x4f>

위와 같이 shareRoom/ 뒤에 랜덤 해쉬값을 동적으로 생성하여 Unique한 방을 생성한다.



The image shows a login form on a dark blue background. At the top, there are two links: 'SIGN IN' (underlined) and 'SIGN UP'. Below these, there is an 'E-MAIL' label followed by a text input field containing 'as00314@nate.com'. Underneath the email field is a 'PASSWORD' label followed by a password input field with four dots and a cursor. At the bottom, there is a blue rounded button labeled 'LOG IN', followed by the text 'FIND PASSWORD' and a link 'NAVER LOGIN'.

- 메인에서 신규 코딩방 생성 버튼을 클릭 -> 방 코딩공유
- 2) 같은 방을 여러 회원끼리 공유하기 위해 크롬 시크릿창을 킨 후, 다른 아이디로 로그인한다.

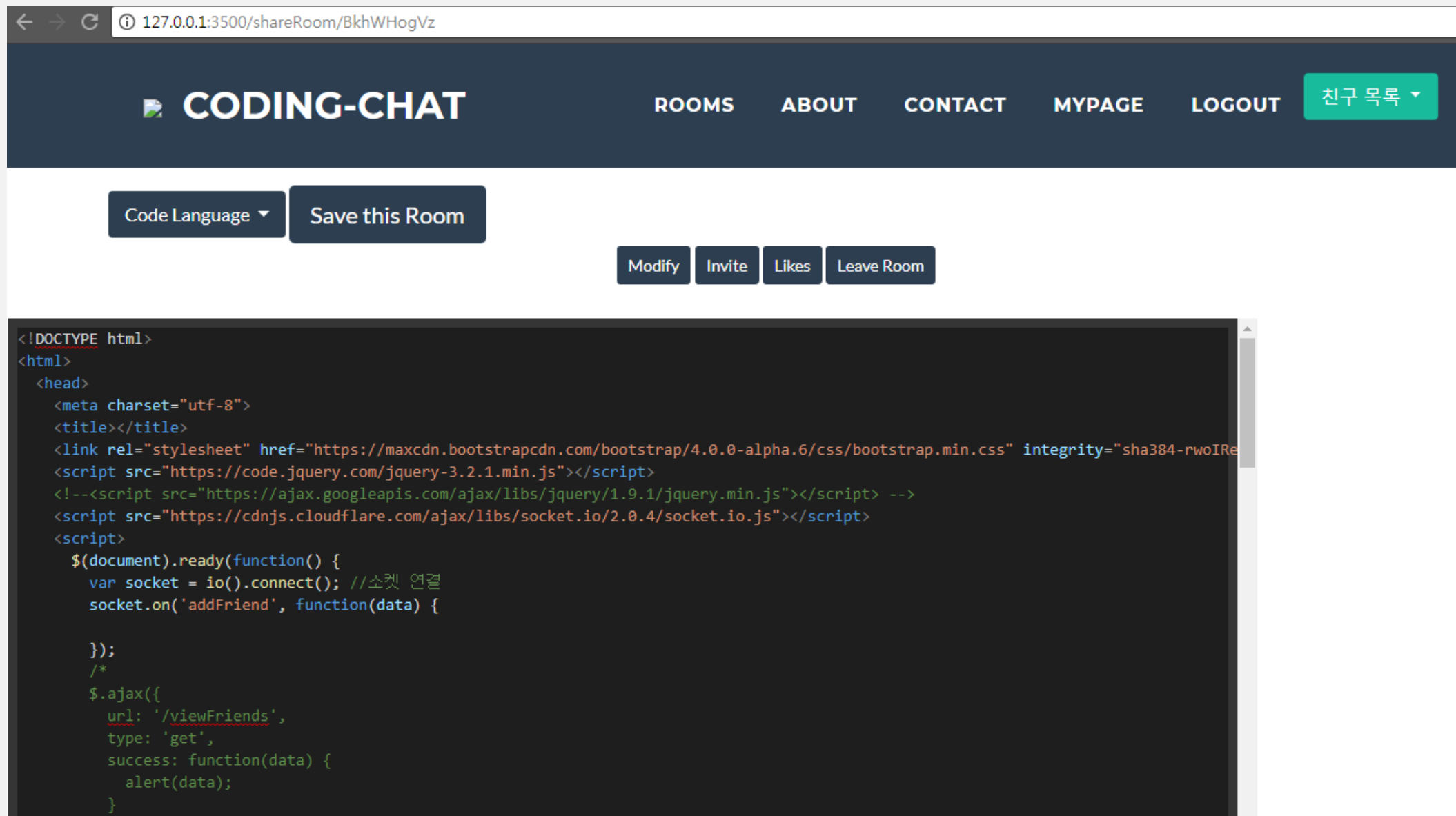
The screenshot displays the 'CODING-CHAT' web application. The browser address bar shows '127.0.0.1:3500/shareRoom/BkhWHogVz'. The navigation bar includes links for ROOMS, ABOUT, CONTACT, MYPAGE, and LOGOUT, along with a '친구 목록' (Friends List) dropdown. Below the navigation bar, there are buttons for 'Code Language' (with a dropdown arrow) and 'Save this Room'. A row of action buttons includes 'Modify', 'Invite', 'Likes', and 'Leave Room'. The main content area features a code editor with the following HTML and JavaScript code:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/css/bootstrap.min.css" integrity="sha384-rwoIRE"
    <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
    <!--<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script> -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/2.0.4/socket.io.js"></script>
    <script>
      $(document).ready(function() {
        var socket = io().connect(); //소켓 연결
        socket.on('addFriend', function(data) {

        });
        /*
        $.ajax({
          url: '/viewFriends',
          type: 'get',
          success: function(data) {
            alert(data);
          }
        });
      });
    </script>
  </head>
</html>
```

On the right side, a vertical list of users is shown with their profile pictures and names: JIHYUN, JANG, SOHYUN, COCORO!, and JISOO-JANG.

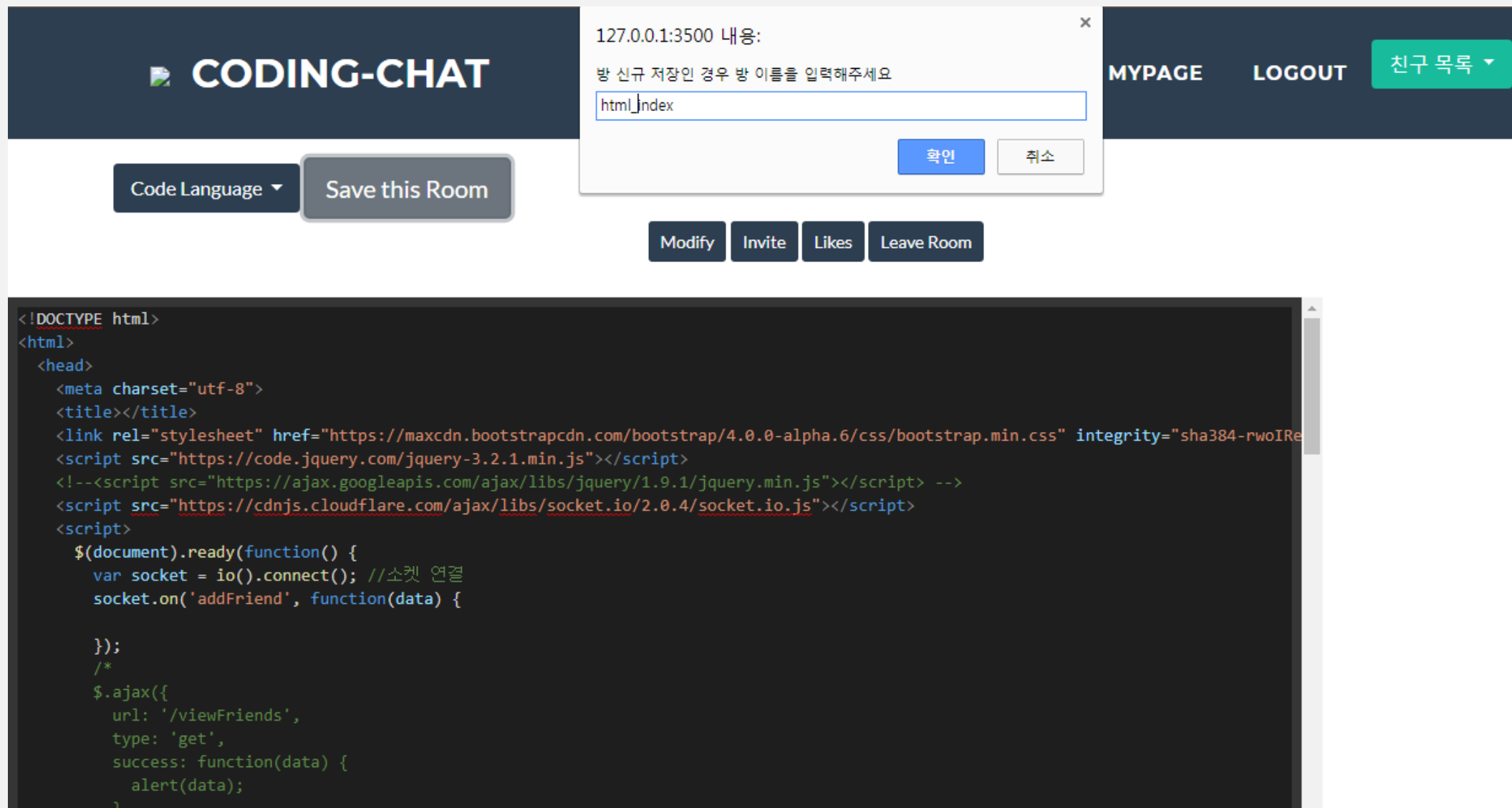
메인에서 신규 코딩방 생성 버튼을 클릭 -> 방 코딩공유
1) 최초 방에 입장한 inspirebj@gmail.com 회원의 방 화면



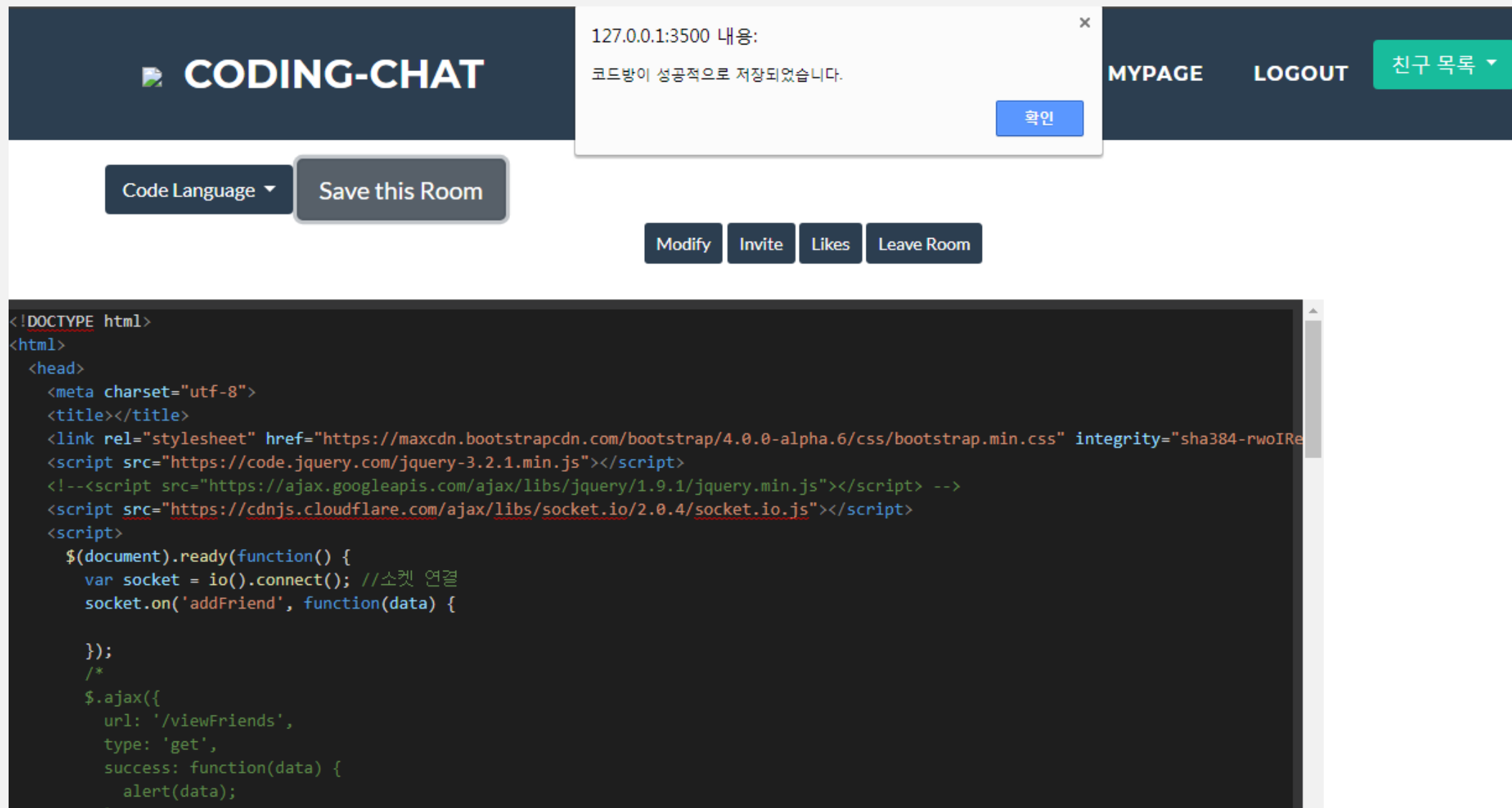
메인에서 신규 코딩방 생성 버튼을 클릭 -> 방 코딩공유

1) 두번째로 방에 입장한 as00314@nate.com 회원의 방 화면

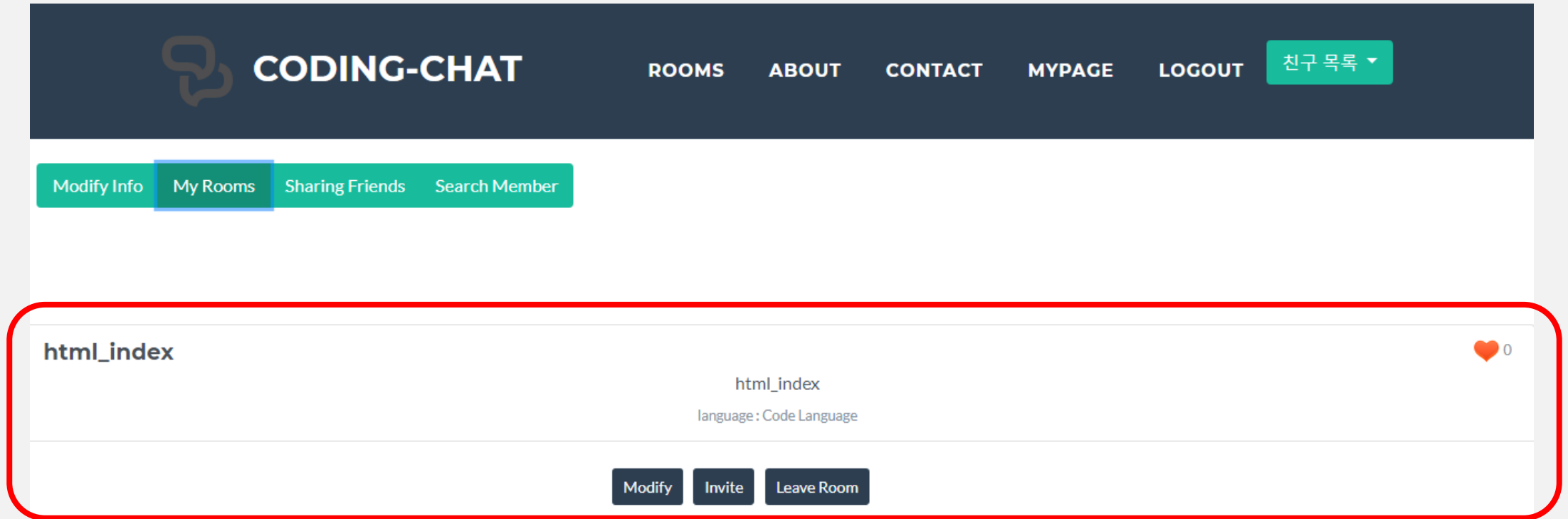
방 입장 시 기존 회원이 작성한 코드를 그대로 볼 수 있고, 실시간으로 내용 수정 시, 방 참여자 모두에게 수정 내용이 반영됨



방 저장 버튼 클릭
방 신규 저장인 경우에는 방 이름을 입력
(DB에서 해당 방이 존재하는지 체크 후, 존재하면 기존 방에서 코드 내용, 코드언어 및 참가자만 수정)



방 저장이 완료됨을 alert창을 통해 성공 메시지를 출력



마이페이지 - My Room
신규로 저장한 방이 리스트에 추가되어 있는 것을 확인할 수 있다.

[ROOMS](#)
[ABOUT](#)
[CONTACT](#)
[MYPAGE](#)
[LOGOUT](#)
[친구 목록 ▾](#)

Code Language ▾

Save this Room

Modify

Invite

Likes

Leave Room

```

<!DOCTYPE html><html> <head> <meta charset="utf-8"> <title></title> <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/css/bootstrap.min.css"
integrity="sha384-rwOJResjU2yc3z8GV/NPeZWA56rSmLldC3R/AZzGRnGxQQKnKkoFVhFQhNUwEyJ" crossorigin="anonymous"> <script src="https://code.jquery.com/jquery-3.2.1.min.js">
</script> <!--<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script> --> <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/2.0.4/socket.io.js"></script>
<script> $(document).ready(function() { var socket = io().connect(); //소켓 연결 socket.on('addFriend', function(data) { }); /* $.ajax({ url: '/viewFriends', type: 'get', success: function(data) {
alert(data); } });*/ $('#friend_list').click(function(e) { $.ajax({ url: '/viewFriends', type: 'get', success: function(data) { for(var i=0 ; i<data.length ; i++) { var html = '<li>' + data[i].profile_image +
'</li>' + '<li><a href="#">' + data[i].id + '</a></li>'; $('<dropdown-menu>').html(html); } }); }); </script> </head> <body> <!-- <a href="/public/login.html" id="login_btn">로그인</a><br>
<a href="/auth/facebook" id="login_btn">페이스북 아이디로 로그인</a><br> <a href="/join/facebook" id="login_btn">페이스북 아이디로 가입</a><br> <a href="/auth/naver"
id="login_btn">네이버 아이디로 로그인</a><br> <a href="/join/naver" id="login_btn">네이버 아이디로 가입</a><br> <a href="/logout" id="login_btn">로그아웃</a><br> <a
href="/public/join.html" id="join_btn">회원가입</a><br> <a href="/findPassword">비밀번호찾기(로그인필요)</a> <hr> <a href="/make_rooms" id="group_btn">방만들기</a><br> <a
href="/public/my_group.html" id="group_btn">참여방 확인</a><br> <a href="/public/finduser.html" id="findUser_btn">회원찾기</a><br> <a href="/viewFriends">친구목록</a> --> <div
id="div1"> <a href="/logout" id="login_btn">로그아웃</a> <a href="/mypage" id="mypage_btn">마이페이지</a> <a href="/public/finduser.html" id="findUser_btn">회원찾기</a><br> <a
href="/make_rooms" id="group_btn">방만들기</a><br> <a href="/public/setImage.html">썸네일 설정</a> <a href="/auth/facebook" id="login_btn">페이스북 아이디로 로그인</a><br> <a
href="/join/facebook" id="login_btn">페이스북 아이디로 가입</a><br> <a href="/auth/naver" id="login_btn">네이버 아이디로 로그인</a><br> <a href="/join/naver" id="login_btn">네이버
아이디로 가입</a><br> <a href="/alert" id="alert_btn"></a> <hr> <div class="dropdown"> <button class="btn btn-primary
dropdown-toggle" id="friend_list" type="button" data-toggle="dropdown">친구 목록 <span class="caret"></span></button> <ul class="dropdown-menu"> </ul> </div> </div> <script
src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js" integrity="sha384-DztdAPBWPRXSA/3eYEEUWRWCy7G5KFbe8ffjk5JAixUYHKkDx6Qin1DkWx51b8rb"
crossorigin="anonymous"></script> <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js" integrity="sha384-
vBWWzLZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyeVke+jo0ieGizqPLForn" crossorigin="anonymous"></script> </body></html>

```

마이페이지 - My Room

신규로 저장한 방을 클릭하면, 기존에 입력하였던 코드 내용이 그대로 저장됨을 확인할 수 있다.



```
function(data) { alert(data); } });*/ $('#friend_list').click(function(e) { $.ajax({ url: '/viewFriends', type: 'get', success: function(data) { for(var i=0 ; i<data.length ; i++) { var html = '<li>' + data[i].profile_image + '</li>' + '<li><a href="#">' + data[i].id + '</a></li>'; $('#dropdown-menu').html(html); } } }); }); </script> </head> <body> <!-- <a href="/public/login.html" id="login_btn">로그인</a><br> <a href="/auth/facebook" id="login_btn">페이스북 아이디로 로그인</a><br> <a href="/join/facebook" id="login_btn">페이스북 아이디로 가입</a><br> <a href="/auth/naver" id="login_btn">네이버 아이디로 로그인</a><br> <a href="/join/naver" id="login_btn">네이버 아이디로 가입</a><br> <a href="/logout" id="login_btn">로그아웃</a><br> <a href="/public/join.html" id="join_btn">회원가입</a><br> <a href="/findPassword" id="login_btn">비밀번호찾기(로그인필요)</a> <hr> <a href="/make_rooms" id="group_btn">방만들기</a><br> <a href="/public/my_group.html" id="group_btn">참여방 확인</a><br> <a href="/public/finduser.html" id="findUser_btn">회원찾기</a><br> <a href="/viewFriends">친구목록</a> --> <div id="div1"> <a href="/logout" id="login_btn">로그아웃</a> <a href="/mypage" id="mypage_btn">마이페이지</a> <a href="/public/finduser.html" id="findUser_btn">회원찾기</a><br> <a href="/make_rooms" id="group_btn">방만들기</a><br> <a href="/public/setImage.html">썸네일 설정</a> <a href="/auth/facebook" id="login_btn">페이스북 아이디로 로그인</a><br> <a href="/join/facebook" id="login_btn">페이스북 아이디로 가입</a><br> <a href="/auth/naver" id="login_btn">네이버 아이디로 로그인</a><br> <a href="/join/naver" id="login_btn">네이버 아이디로 가입</a><br> <a href="/alert" id="alert_btn"></a> <hr> <div class="dropdown"> <button class="btn btn-primary dropdown-toggle" id="friend_list" type="button" data-toggle="dropdown">친구 목록 <span class="caret"></span></button> <ul class="dropdown-menu"> </ul> </div> </div> <script src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js" integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KfBe8ffjk5JAIXUYHKkDx6Qin1DkWx51bBrb" crossorigin="anonymous"></script> <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js" integrity="sha384-vBWwZlZJ8ea9aCX4pEW3rVHjgt7zpkNpZk+02D9phzyeVKE+jo0ieGizqPLForm" crossorigin="anonymous"></script> </body></html>
```

html_index Participants Info



inspirebj@gmail.com

as00314@nate.com

마이페이지 - My Room

신규로 저장한 방을 클릭하면, 하단에 방을 저장한 참여자 회원 정보를 볼 수 있다.

CODING-CHAT

ROOMS

ABOUT

CONTACT

MYPAGE

LOGOUT

친구 목록 ▾

Modify

Invite

Likes

Leave Room

```

var socket = io().connect(); //소켓 연결 socket.on('addFriend', function(data) { }); /* $.ajax({ url: '/viewFriends', type: 'get', success:
function(data) { alert(data); } });*/ $('#friend_list').click(function(e) { $.ajax({ url: '/viewFriends', type: 'get', success: function(data) { for(var i=0 ;
i<data.length ; i++) { var html = '<li>' + data[i].profile_image + '</li>' + '<li><a href="#">' + data[i].id + '</a></li>'; $('.dropdown-
menu').html(html); } } }); }); </script> </head> <body> <!-- <a href="/public/login.html" id="login_btn">로그인</a><br> <a
href="/auth/facebook" id="login_btn">페이스북 아이디로 로그인</a><br> <a href="/join/facebook" id="login_btn">페이스북 아이디로 가입
</a><br> <a href="/auth/naver" id="login_btn">네이버 아이디로 로그인</a><br> <a href="/join/naver" id="login_btn">네이버 아이디로 가
입</a><br> <a href="/logout" id="login_btn">로그아웃</a><br> <a href="/public/join.html" id="join_btn">회원가입</a><br> <a
href="/findPassword">비밀번호찾기(로그인필요)</a> <hr> <a href="/make_rooms" id="group_btn">방만들기</a><br> <a
href="/public/my_group.html" id="group_btn">참여방 확인</a><br> <a href="/public/finduser.html" id="findUser_btn">회원찾기</a><br>
<a href="/viewFriends">친구목록</a> --> <div id="div1"> <a href="/logout" id="login_btn">로그아웃</a> <a href="/mypage"
id="mypage_btn">마이페이지</a> <a href="/public/finduser.html" id="findUser_btn">회원찾기</a><br> <a href="/make_rooms"
id="group_btn">방만들기</a><br> <a href="/public/setImage.html">썸네일 설정</a> <a href="/auth/facebook" id="login_btn">페이스북
아이디로 로그인</a><br> <a href="/join/facebook" id="login_btn">페이스북 아이디로 가입</a><br> <a href="/auth/naver"
id="login_btn">네이버 아이디로 로그인</a><br> <a href="/join/naver" id="login_btn">네이버 아이디로 가입</a><br> <a href="/alert"
id="alert_btn"></a> <hr> <div class="dropdown"> <button class="btn btn-primary
dropdown-toggle" id="friend_list" type="button" data-toggle="dropdown">친구 목록 <span class="caret"></span></button> <ul
class="dropdown-menu"> </ul> </div> </div> <script src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js"
integrity="sha384-DztdAPBWPRXSA/3eYEEUWRWcY7G5KfBe8fjk5JAIXUYHKkDx6Qin1DkWx51bBrb" crossorigin="anonymous"> </script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js" integrity="sha384-
vBWWzlZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyeVkE+jo0ieGizqPLForn" crossorigin="anonymous"> </script> </body></html>
modifying.....

```

마이페이지 - My Room

신규로 저장한 방을 클릭하여 입장 후, 코드 내용을 수정해본다.

127.0.0.1:3500 내용:

방 신규 저장인 경우 방 이름을 입력해주세요

확인 취소

html Save this Room

Modify Invite Likes Leave Room

```

</script> </script> <script src="https://cdn.jsdelivr.net/npm/jquery@1.12.4/dist/jquery.min.js"></script> <script src="https://cdn.jsdelivr.net/npm/tether@1.4.0/dist/tether.min.js"></script> <script src="https://cdn.jsdelivr.net/npm/tether@1.4.0/dist/tether.min.js"></script>
var socket = io().connect(); //소켓 연결 socket.on('addFriend', function(data) {}); /* $.ajax({ url: '/viewFriends', type: 'get', success:
function(data) { alert(data); } });*/ $('#friend_list').click(function(e) { $.ajax({ url: '/viewFriends', type: 'get', success: function(data) { for(var i=0 ;
i<data.length ; i++) { var html = '<li>' + data[i].profile_image + '</li>' + '<li><a href="#">' + data[i].id + '</a></li>'; $('#dropdown-
menu').html(html); } } }); }); </script> </head> <body> <!-- <a href="/public/login.html" id="login_btn">로그인</a><br> <a
href="/auth/facebook" id="login_btn">페이스북 아이디로 로그인</a><br> <a href="/join/facebook" id="login_btn">페이스북 아이디로 가입
</a><br> <a href="/auth/naver" id="login_btn">네이버 아이디로 로그인</a><br> <a href="/join/naver" id="login_btn">네이버 아이디로 가
입</a><br> <a href="/logout" id="login_btn">로그아웃</a><br> <a href="/public/join.html" id="join_btn">회원가입</a><br> <a
href="/findPassword">비밀번호찾기(로그인필요)</a> <hr> <a href="/make_rooms" id="group_btn">방만들기</a><br> <a
href="/public/my_group.html" id="group_btn">참여방 확인</a><br> <a href="/public/finduser.html" id="findUser_btn">회원찾기</a><br>
<a href="/viewFriends">친구목록</a> --> <div id="div1"> <a href="/logout" id="login_btn">로그아웃</a> <a href="/mypage"
id="mypage_btn">마이페이지</a> <a href="/public/finduser.html" id="findUser_btn">회원찾기</a><br> <a href="/make_rooms"
id="group_btn">방만들기</a><br> <a href="/public/setImage.html">썸네일 설정</a> <a href="/auth/facebook" id="login_btn">페이스북
아이디로 로그인</a><br> <a href="/join/facebook" id="login_btn">페이스북 아이디로 가입</a><br> <a href="/auth/naver"
id="login_btn">네이버 아이디로 로그인</a><br> <a href="/join/naver" id="login_btn">네이버 아이디로 가입</a><br> <a href="/alert"
id="alert_btn"></a> <hr> <div class="dropdown"> <button class="btn btn-primary
dropdown-toggle" id="friend_list" type="button" data-toggle="dropdown">친구 목록 <span class="caret"></span></button> <ul
class="dropdown-menu"> </ul> </div> </div> <script src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js"
integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KfBe8fjk5JAiXUYHKkDx6Qin1DkWx51bBrb" crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js" integrity="sha384-
vBWwZlZl8ea9aCX4nEW3rVHqit7znkNpZk+02D9nbhzeV/kE+io0ieGizgPIEom" crossorigin="anonymous"></script> </body> </html>

```

마이페이지 - My Room

신규로 저장한 방을 클릭하여 입장 후, 코드 내용을 수정한 후, 다시 저장
(신규 저장이 아니므로, 코드 내용과, 코드 언어, 참여자 정보만 체크 후 DB에서 update 된다)



```

</script> </script> <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/2.0.1/socket.io.js"></script> <script> $(document).ready(function() {
var socket = io().connect(); //소켓 연결 socket.on('addFriend', function(data) { }); /* $.ajax({ url: '/viewFriends', type: 'get', success:
function(data) { alert(data); } });*/ $('#friend_list').click(function(e) { $.ajax({ url: '/viewFriends', type: 'get', success: function(data) { for(var i=0 ;
i<data.length ; i++) { var html = '<li>' + data[i].profile_image + '</li>' + '<li><a href="#">' + data[i].id + '</a></li>'; $('#dropdown-
menu').html(html); } } }); }); </script> </head> <body> <!-- <a href="/public/login.html" id="login_btn">로그인</a><br> <a
href="/auth/facebook" id="login_btn">페이스북 아이디로 로그인</a><br> <a href="/join/facebook" id="login_btn">페이스북 아이디로 가입
</a><br> <a href="/auth/naver" id="login_btn">네이버 아이디로 로그인</a><br> <a href="/join/naver" id="login_btn">네이버 아이디로 가
입</a><br> <a href="/logout" id="login_btn">로그아웃</a><br> <a href="/public/join.html" id="join_btn">회원가입</a><br> <a
href="/findPassword">비밀번호찾기(로그인필요)</a> <hr> <a href="/make_rooms" id="group_btn">방만들기</a><br> <a
href="/public/my_group.html" id="group_btn">참여방 확인</a><br> <a href="/public/finduser.html" id="findUser_btn">회원찾기</a><br>
<a href="/viewFriends">친구목록</a> --> <div id="div1"> <a href="/logout" id="login_btn">로그아웃</a> <a href="/mypage"
id="mypage_btn">마이페이지</a> <a href="/public/finduser.html" id="findUser_btn">회원찾기</a><br> <a href="/make_rooms"
id="group_btn">방만들기</a><br> <a href="/public/setImage.html">썸네일 설정</a> <a href="/auth/facebook" id="login_btn">페이스북
아이디로 로그인</a><br> <a href="/join/facebook" id="login_btn">페이스북 아이디로 가입</a><br> <a href="/auth/naver"
id="login_btn">네이버 아이디로 로그인</a><br> <a href="/join/naver" id="login_btn">네이버 아이디로 가입</a><br> <a href="/alert"
id="alert_btn"></a> <hr> <div class="dropdown"> <button class="btn btn-primary
dropdown-toggle" id="friend_list" type="button" data-toggle="dropdown">친구 목록 <span class="caret"></span></button> <ul
class="dropdown-menu"> </ul> </div> </div> <script src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js"
integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KfBe8fjk5JAiXUYHKkDx6Qin1DkWx51bBrb" crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js" integrity="sha384-
vBWw-l718e9e9CY4pEW3A/Hijjt7zpkNpZk+02D9phzyeVKE+jo0ieGizqPLForn" crossorigin="anonymous"></script> </body>
</html>

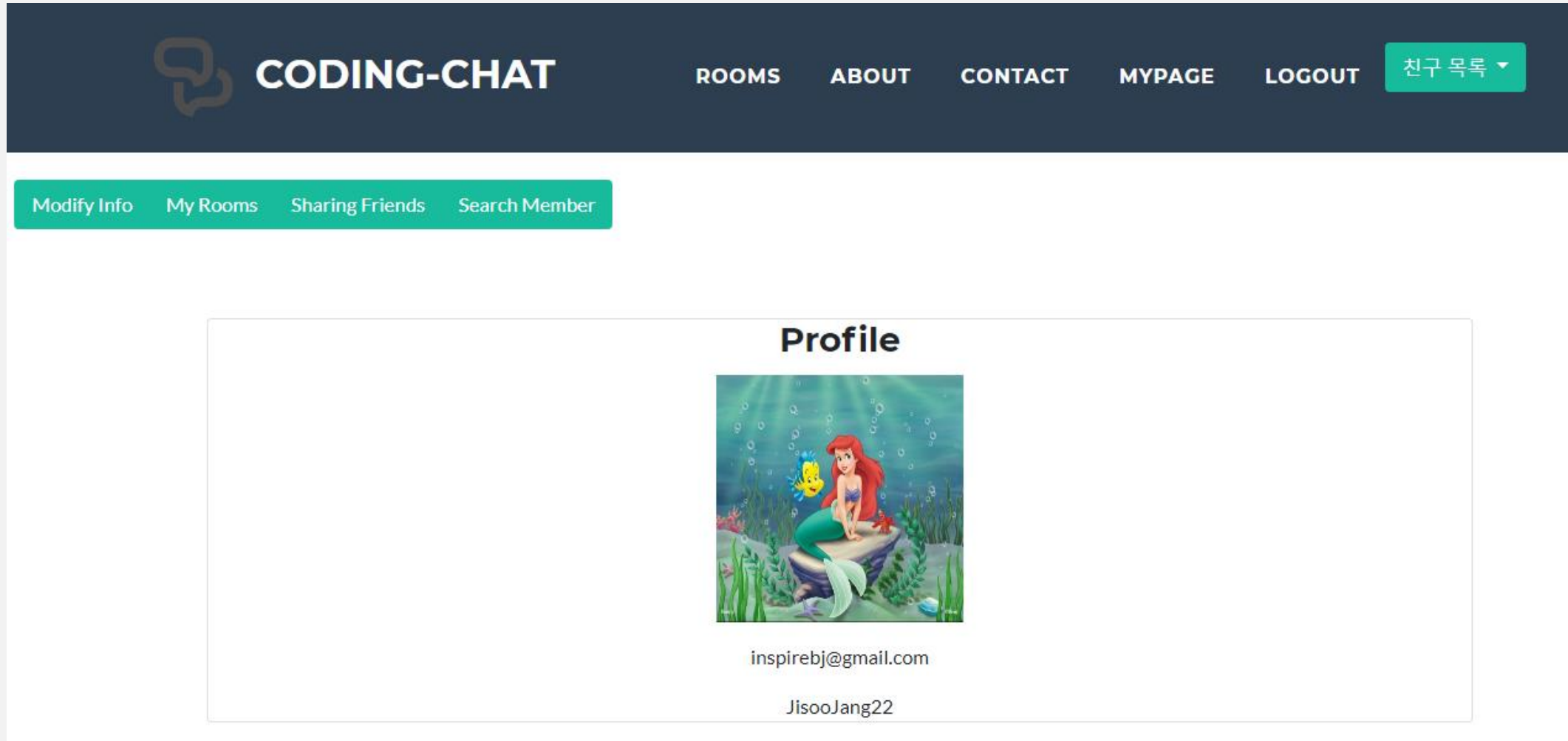
```

html_index Participants Info




마이페이지 - My Room


방 수정을 완료한 후, 다시 방 리스트에서 해당 방에 입장하면
코드 수정 내용이 반영되어있는 것을 확인할 수 있다.



상단에서 MYPAGE 버튼을 클릭하면, 세션에 로그인된 회원 아이디, 닉네임, 프로필 이미지를 보여준다.
MYPAGE 화면에서는 회원정보수정, 참여 방 정보, 친구 목록, 멤버 찾기 기능이 제공된다.

 **CODING-CHAT**

ROOMS ABOUT CONTACT MYPAGE LOGOUT 친구 목록 ▾


Profile_Thumbnail
(권장사이즈 : 200 * 200)
파일 선택 선택된 파일 없음

ID (Email):
inspirebj@gmail.com

Password:

Password (Confirm):

Nickname:

Submit

회원정보 수정 화면에서 폼에 수정할 내용을 입력한다.
폼에서는 ID와 닉네임은 기존 회원 정보를 그대로 폼에 입력해둔다.
폼에서 프로필이미지 파일을 별도로 선택하지 않으면, 기존의 프로필 사진을 그대로 사용한다.



CODING-CHAT

ROOMS

ABOUT

CONTACT

MYPAGE

LOGOUT

친구 목록 ▾

Modify Info

My Rooms

Sharing Friends

Search Member

Profile



inspirebj@gmail.com

JisooJang_modified

회원 정보 수정 후, MYPAGE 화면
폼 변경 내용이 적용된 것을 확인할 수 있다.

127.0.0.1:3500 내용:

초대할 회원의 이메일 주소를 입력해주세요!

ppuagirls@naver.com

확인 취소

Modify Invite Likes Leave Room

```
<script> $('#logout').click(function(e) { $.ajax({ url: '/logout', type: 'get', success: function(data) { if(data.alert_message == "로그아웃 완료") { alert('로그아웃이 완료되었습니다.');
```

특정 방에 입장 후, 방 초대 버튼 클릭
초대할 회원의 이메일 주소를 입력

이 때, 회원 DB에 해당 이메일 아이디가 있는지 체크 후, 존재하면 메일 발송

127.0.0.1:3500 내용:

사용자에게 성공적으로 초대메일을 전송하였습니다.

확인

```
<script> $('#logout').click(function(e) { $.ajax({ url: '/logout', type: 'get', success: function(data) { if(data.alert_message == "로그아웃 완료") { alert('로그아웃이 완료되었습니다.');

ROOMS



ABOUT



CONTACT



MYPAGE



LOGOUT



친구 목록



Modify



Invite



Likes



Leave Room



</div> </div> </nav>ㅋㅋㅋㅎㅎㅎ하얏르저르ㅋㅋㅋㄴ르ㄷ르ㄷ지jjiehfrfrfwefrferf


```

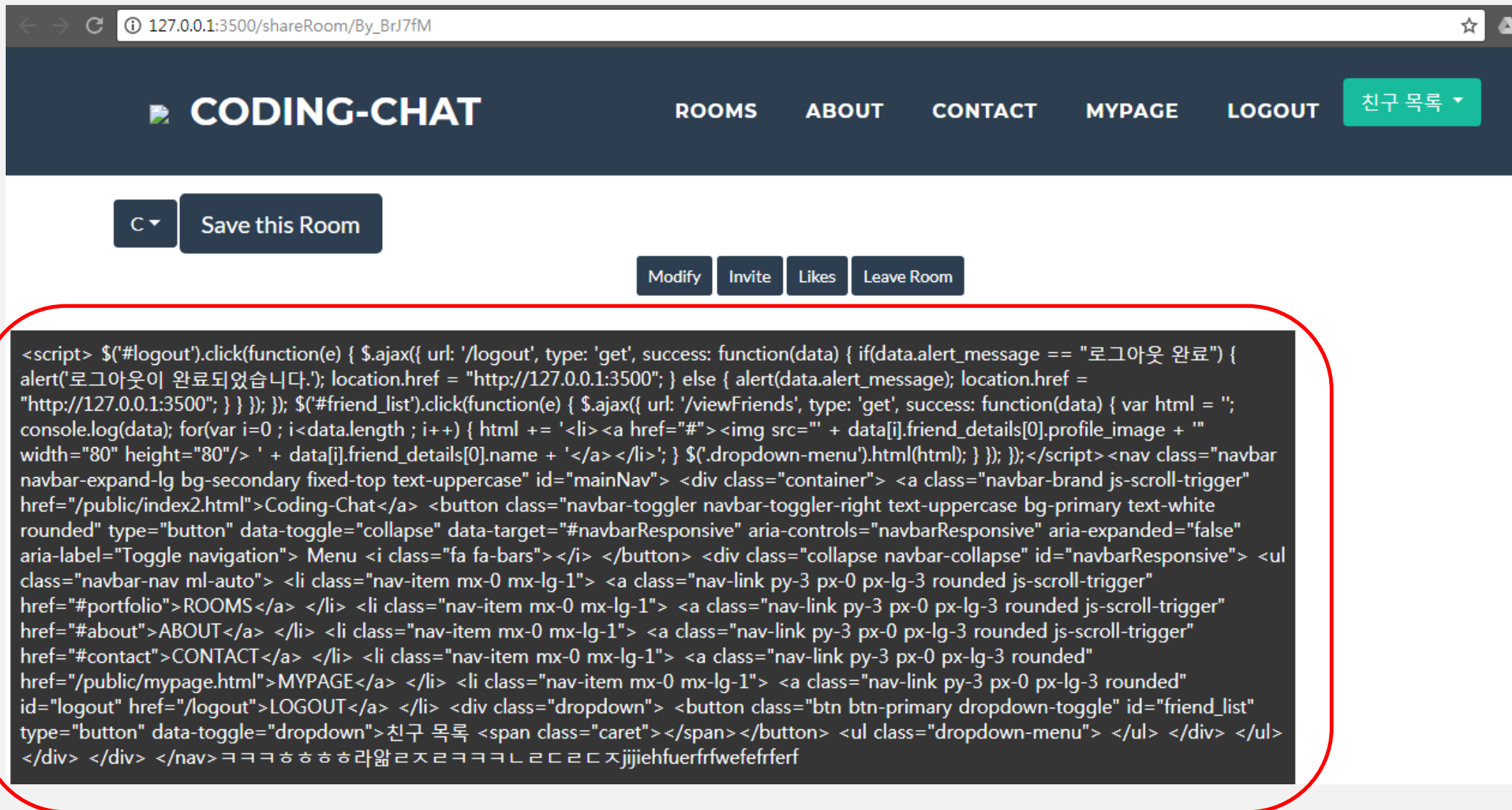
특정 방에 입장 후, 방 초대 버튼 클릭
초대 메일이 성공적으로 전송되었으면, 성공 메시지를 alert창을 통해 출력한다.



특정 방 초대 메일 수신 화면

입력한 이메일로 방 이름과 방 설명 내용을 담은 메일을 전송한다.

하단에 방 입장하기 버튼을 클릭하면, 각 방의 고유 URL로 연결하여 같은 방에서 코드 실시간 공유가 가능하다.



127.0.0.1:3500/shareRoom/By_BrJ7fM

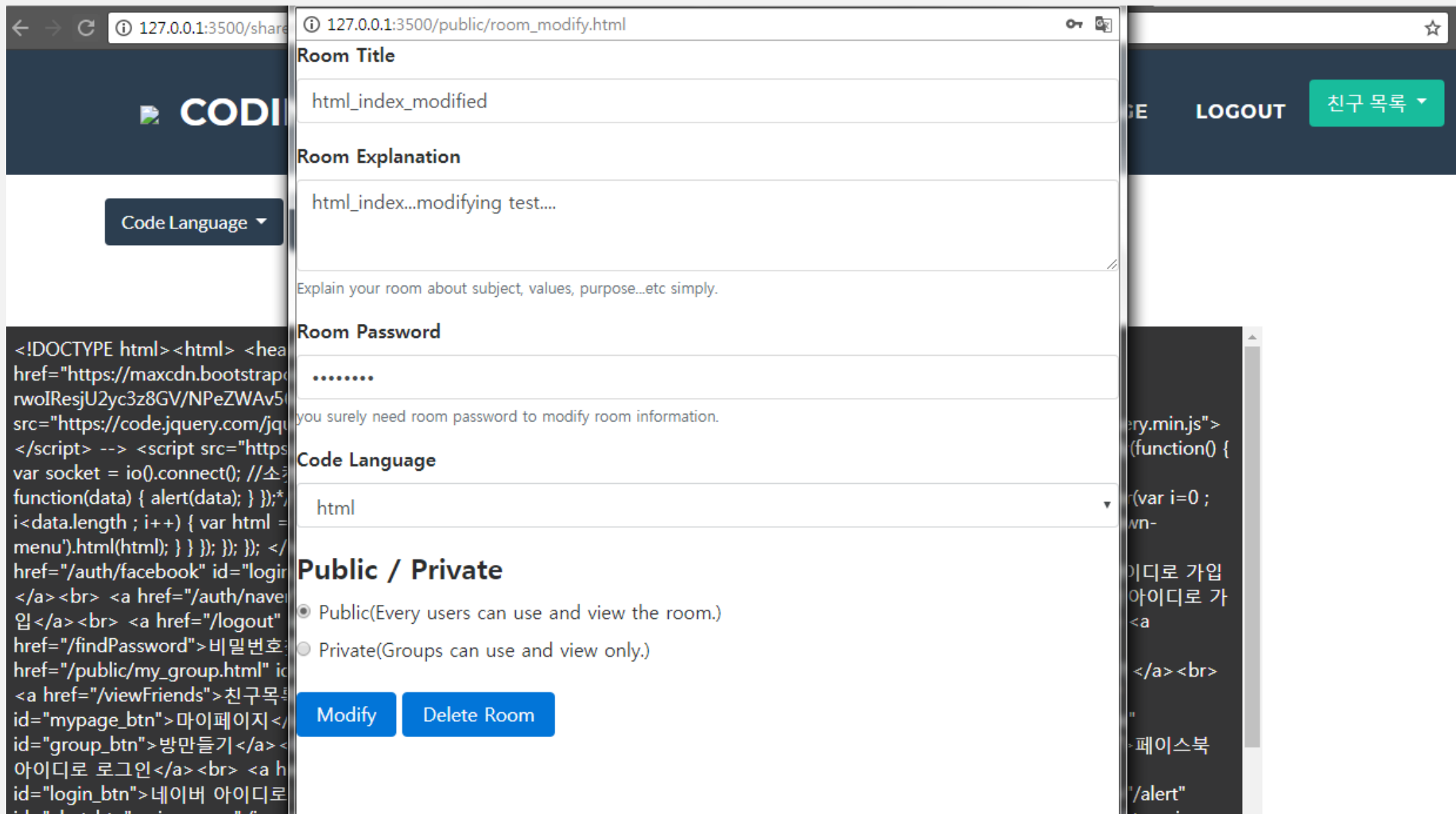
CODING-CHAT ROOMS ABOUT CONTACT MYPAGE LOGOUT 친구 목록 ▾

C ▾ Save this Room

Modify Invite Likes Leave Room

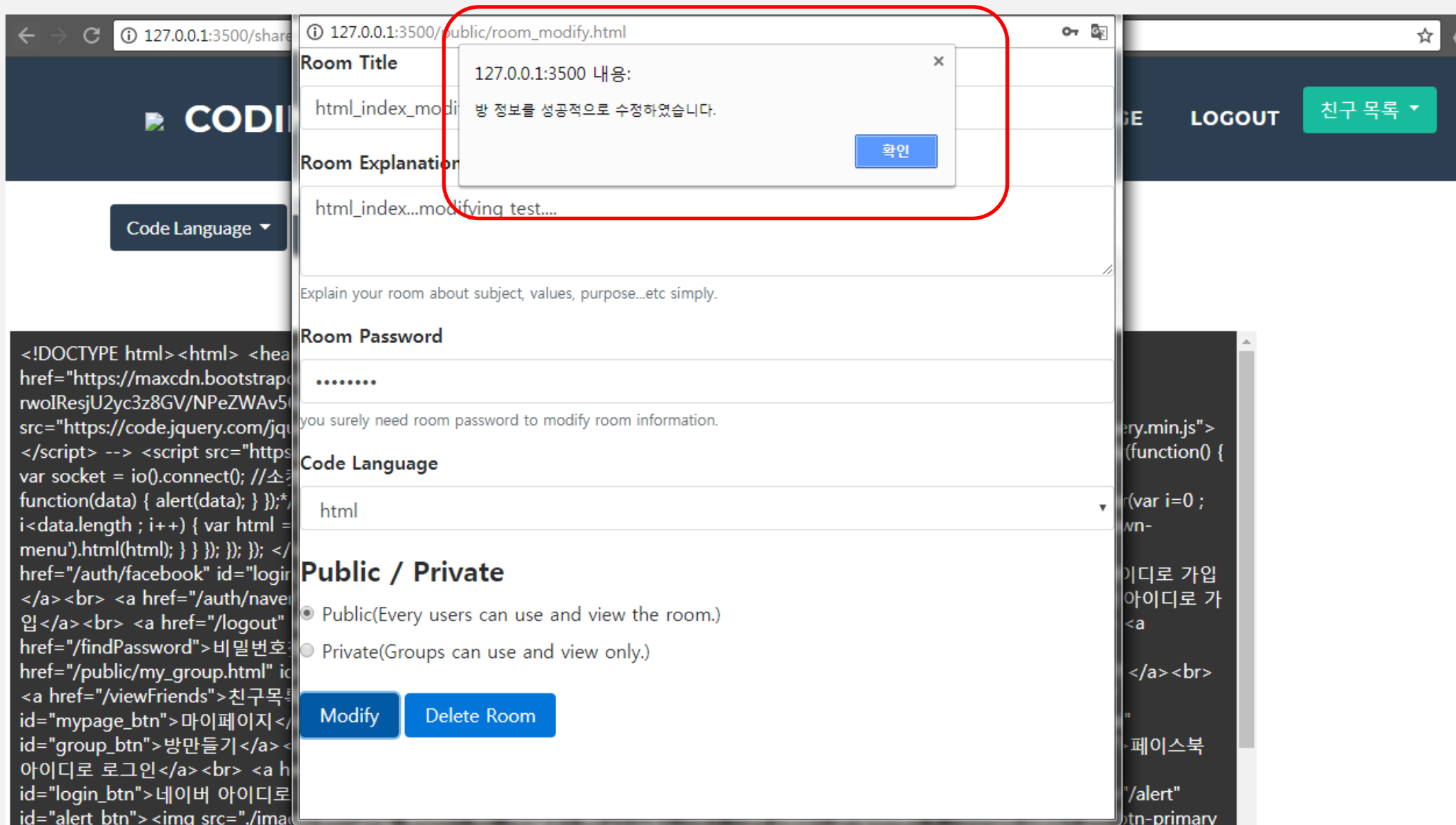
```
<script> $('#logout').click(function(e) { $.ajax({ url: '/logout', type: 'get', success: function(data) { if(data.alert_message == "로그아웃 완료") { alert('로그아웃이 완료되었습니다.');
```

특정 방 초대 메일 에서 방 입장 버튼 클릭 시 - 같은 방 코드 공유 가능



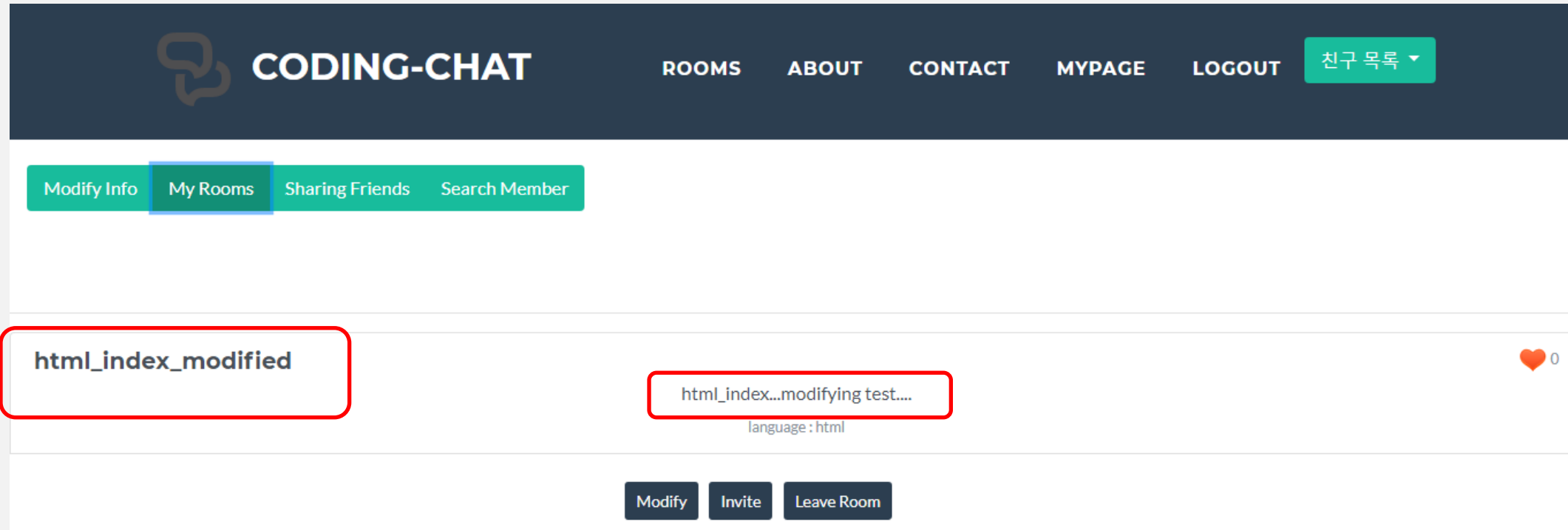
특정 방 입장 후, 방 정보 수정 버튼 클릭

방 정보를 수정할 수 있는 폼을 새 창으로 띄운다. (방 이름, 방 설명, 방 비밀번호, 코드 언어, 공유 여부)

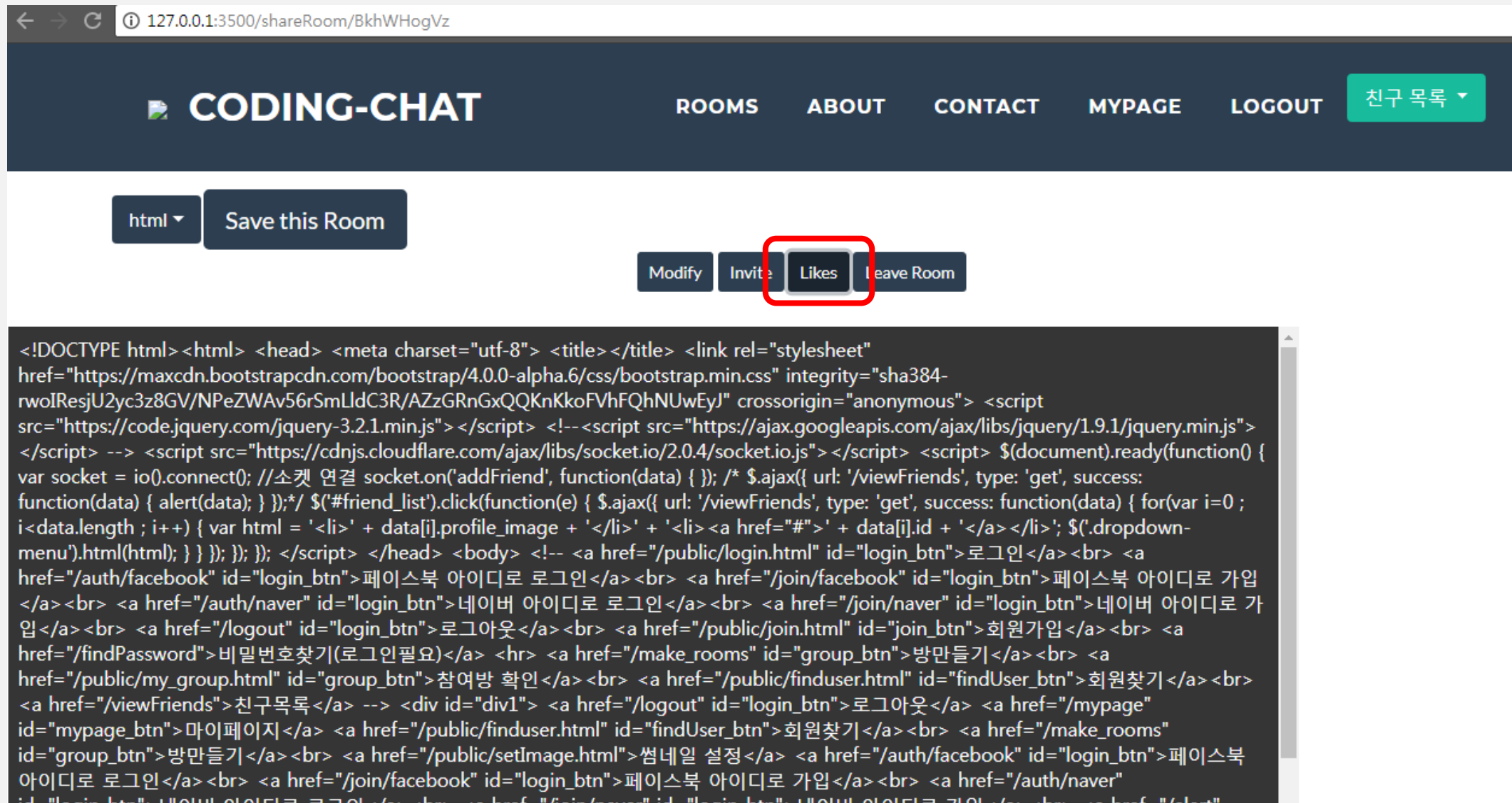


특정 방 입장 후, 방 정보 수정 버튼 클릭

방 수정 버튼을 클릭하면, 수정 내용을 Rooms DB에 반영하고, 성공 여부를 alert 메시지로 출력한다.

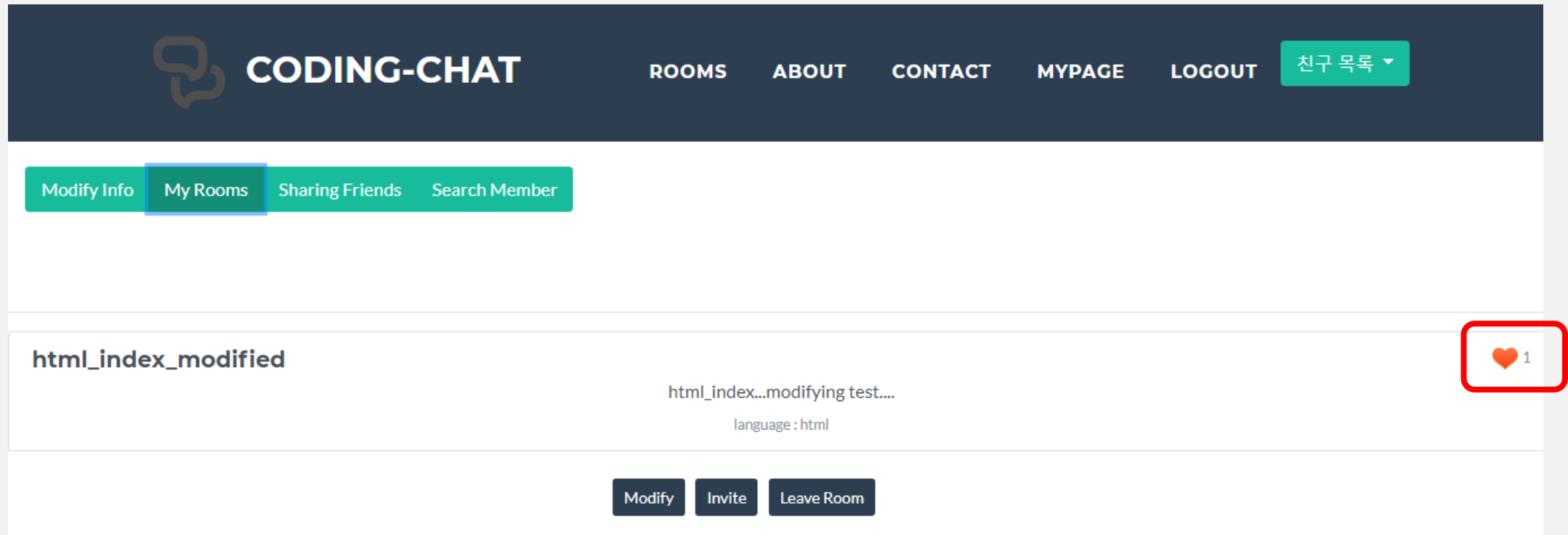


방 정보 수정 완료 후,
MYPAGE - My ROOMS로 들어가서 참여 방 정보를 보면, 수정 내용이 반영된 것을 확인할 수 있다.




특정 방에 입장 후, 좋아요 버튼을 클릭한다.

서버에서는 현재 로그인된 회원이 해당 방에 대해서 좋아요를 눌렀는지에 대한 기록을 DB에서 체크한 후
좋아요를 증가시킬지 감소시킬지 결정한다.




특정 방에 입장 후, 좋아요 버튼을 클릭한다.
서버에서는 현재 로그인된 회원이 해당 방에 대해서 좋아요를 눌렀는지에 대한 기록을 DB에서 체크한 후
좋아요를 증가시킬지 감소시킬지 결정한다.

이 신규방에 대한 좋아요 기록이 없으므로, 좋아요가 1 증가된 것을 확인할 수 있다.

 **CODING-CHAT** ROOMS ABOUT CONTACT **MYPAGE** LOGOUT 친구 목록 ▾


Modify Info My Rooms **Sharing Friends** Search Member



ppuagirls@naver.com

Jihyun Jang

참여 방 확인 1:1대화




as00314@nate.com

Sohyun

참여 방 확인 1:1대화

MYPAGE - Sharing Friends 버튼을 클릭하면 친구 추가된 친구목록을 확인할 수 있다.

 CODING-CHAT

ROOMS ABOUT CONTACT MYPAGE LOGOUT 친구 목록 ▾

Modify Info My Rooms Sharing Friends Search Member

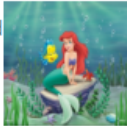
이메일

inspirebj@gmail.com

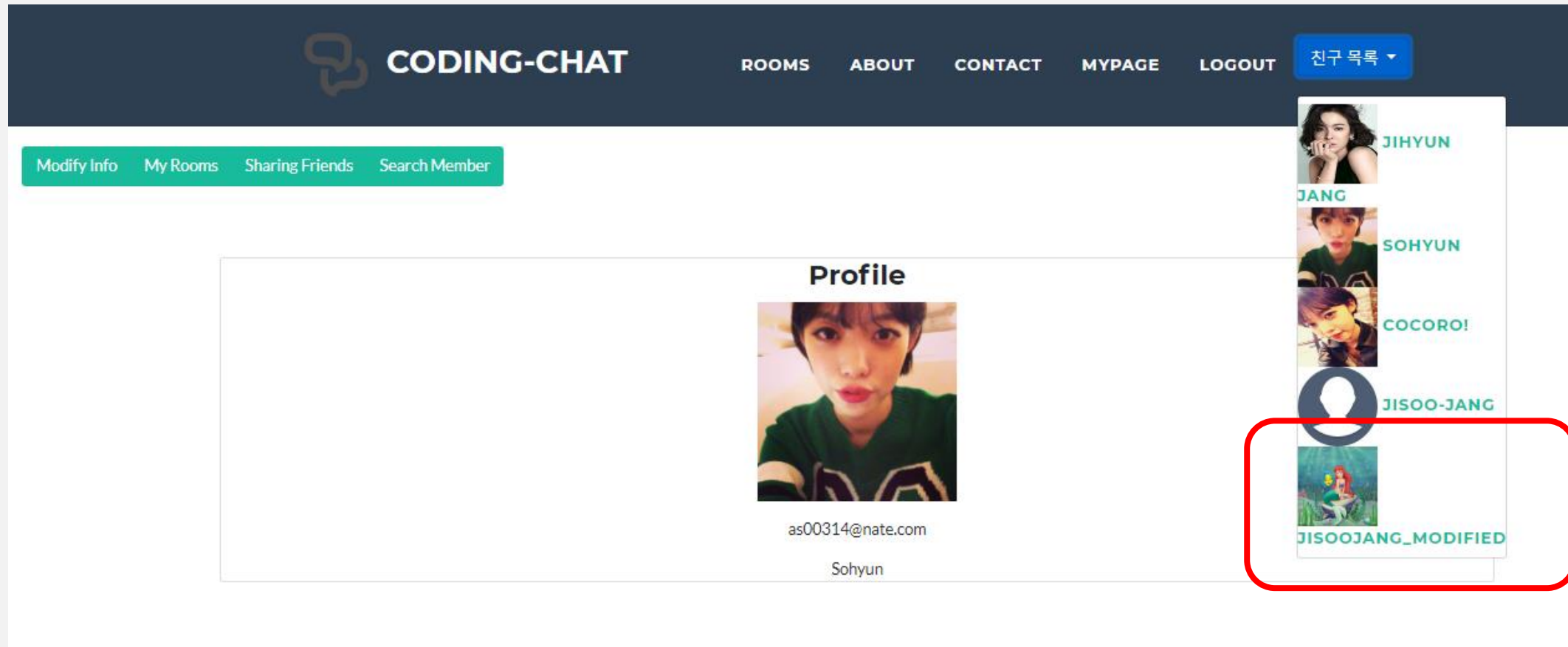
search

1. id - inspirebj@gmail.com
nickName - JisooJang_modified

친구요청



MYPAGE - Search Member 버튼을 클릭하면 이메일 또는 닉네임으로 가입된 회원정보를 찾을 수 있다.
검색버튼 클릭 시, 하단에 ajax로 검색된 회원정보를 추가하여 id, 닉네임, 프로필이미지를 보여주고
친구요청을 할 수 있도록 구현하였다.



MYPAGE - Search Member 에서 친구 추가 버튼 클릭 시,
친구 목록 화면에서 신규로 해당 회원이 추가 된 것을 확인할 수 있다.