
How about Potato

목차

01. HAP

개발환경

설계

구현내용

마무리

개발환경



OS - Windows 7



DB Client - Oracle 19.1.0



IDE – sts 3.9.10



Build Tool - Maven



DB Modeling Tool - eXERD



front making - vs code



Image Making - mangoboard



GitHub.com

설계 eXERD 구조

논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
공지번호	notice_num	N/A	NUMBER	N·N
회원번호	member_num	N/A	NUMBER	N·N
제목	title	N/A	VARCHAR(50)	N·N
내용	content	N/A	VARCHAR(4000)	N·N
조회수	viewcnt	N/A	NUMBER(5)	N·N
비밀번호	passwd	N/A	VARCHAR(20)	N·N
날짜	wdate	N/A	DATE	N·N
파일 이름	filename	N/A	VARCHAR(100)	NULL
파일 사이즈	filesize	N/A	NUMBER(5)	NULL
상단공지	noti	N/A	VARCHAR(1)	NULL

논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
회원번호	member_num	N/A	NUMBER	N·N
아이디	id	N/A	VARCHAR(20)	N·N
이름	name	N/A	VARCHAR(20)	N·N
비밀번호	passwd	N/A	VARCHAR(20)	N·N
이메일	email	N/A	VARCHAR(30)	N·N
핸드폰번호	phone	N/A	VARCHAR(15)	N·N
생일	birth	N/A	VARCHAR(20)	N·N
등급	grade	N/A	VARCHAR(1)	NULL
별명	nick	N/A	VARCHAR(10)	N·N
배경사진	backfile	N/A	VARCHAR(100)	NULL
프로필사진	fname	N/A	VARCHAR(100)	NULL

논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
질문번호	qna_num	N/A	NUMBER	N·N
회원번호	member_num	N/A	NUMBER	N·N
제목	title	N/A	VARCHAR(50)	NULL
내용	content	N/A	VARCHAR(4000)	N·N
조회수	viewcnt	N/A	NUMBER(5)	N·N
비밀번호	passwd	N/A	VARCHAR(20)	N·N
날짜	wdate	N/A	DATE	N·N
파일 이름	filename	N/A	VARCHAR(100)	NULL
파일 사이즈	filesize	N/A	NUMBER(50)	NULL
상단공지	noti	N/A	VARCHAR(1)	NULL
부모번호	ref	N/A	NUMBER(7)	N·N
답변갯수	ansnum	N/A	NUMBER(5)	N·N
답글들여쓰기	indent	N/A	NUMBER(2)	N·N
부모글번호구분	refnum	N/A	NUMBER(7)	N·N

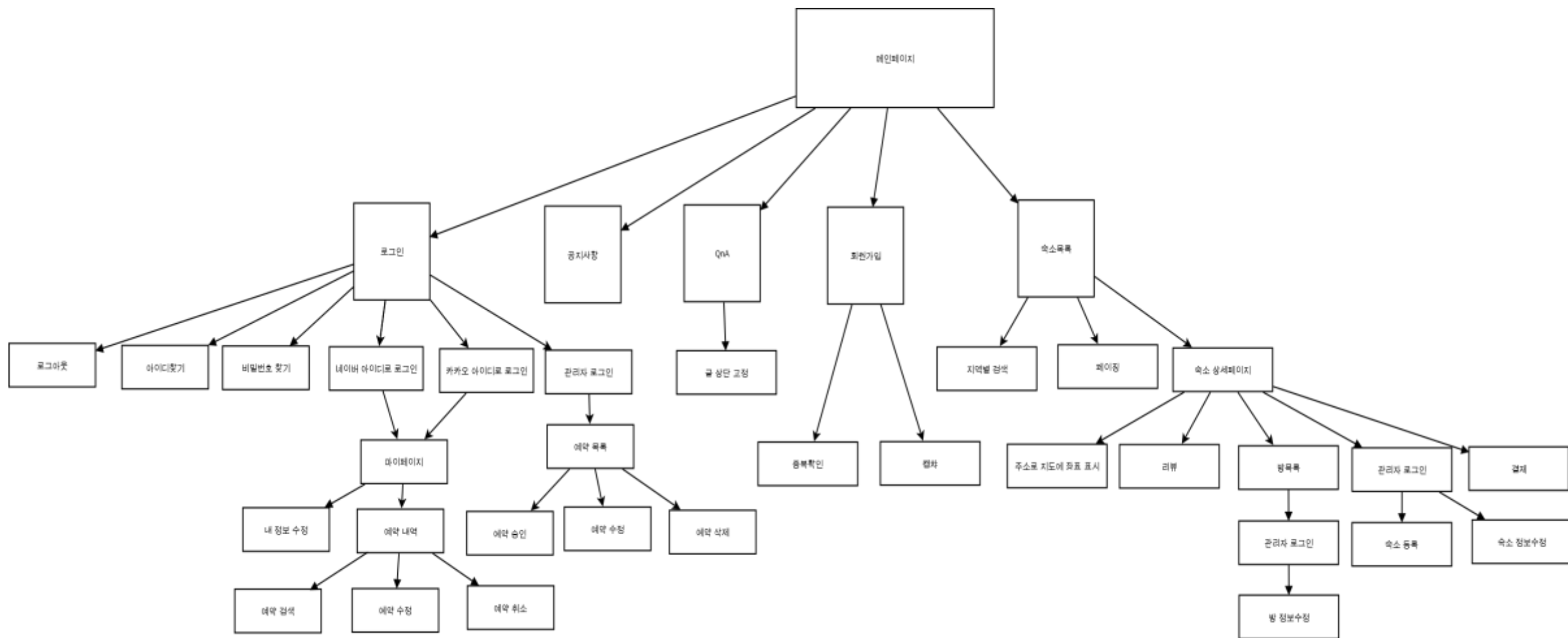
논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
방번호	board_num	N/A	NUMBER	N·N
이름	name	N/A	VARCHAR(20)	N·N
주소	address1	N/A	VARCHAR(100)	N·N
상세주소	address2	N/A	VARCHAR(100)	N·N
파일명	filename	N/A	VARCHAR(100)	N·N
계좌	account	N/A	VARCHAR(30)	N·N
내용	content	N/A	VARCHAR(4000)	N·N
조회수	cnt	N/A	NUMBER(5)	N·N
우편번호	zipcode	N/A	VARCHAR(100)	N·N

논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
정보번호	room_num	N/A	NUMBER	N·N
방번호	board_num	N/A	NUMBER	N·N
사진	filename	N/A	VARCHAR(100)	N·N
가격	price	N/A	NUMBER(10)	N·N
이름	name	N/A	VARCHAR(20)	N·N
내용	content	N/A	VARCHAR(4000)	N·N
인원	max	N/A	NUMBER(1)	N·N

논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
예약번호	res_num	N/A	NUMBER	N·N
정보번호	room_num	N/A	NUMBER	N·N
방번호	board_num	N/A	NUMBER	N·N
회원번호	member_num	N/A	NUMBER	N·N
지정번호	rannum	N/A	NUMBER(8)	N·N
이름	name	N/A	VARCHAR(20)	N·N
핸드폰번호	phone	N/A	VARCHAR(15)	N·N
내용	content	N/A	VARCHAR(4000)	NULL
입실일	res1	N/A	DATE	N·N
퇴실일	res2	N/A	DATE	N·N
인원	people	N/A	NUMBER(1)	NULL
승인	acc	N/A	VARCHAR(1)	NULL

설계

알고리즘



구현내용 1-1 회원가입

- ID중복확인
- E-mail중복확인
- reCAPTCHA기능

HAP 회원 가입 양식

(* 은 필수입력 항목입니다.)

사진

파일 선택

선택된 파일 없음

*아이디

Enter 아이디

ID 중복확인

*이름

Enter 이름

*비밀번호

Enter 비밀번호

*비밀번호확인

Enter 비밀번호 확인

*생년월일

Enter 생년월일

*전화번호

Enter 전화번호


*E-mail

Enter E-mail

E-mail 중복확인

*로봇 확인

☐ 로봇이 아닙니다.


reCAPTCHA
개인정보 보호 - 약관

리캡차 확인

등록

취소

*아이디

user1

ID 중복확인

user1은(는) 중복된 아이디입니다. 다른 아이디를 입력해주세요.


*E-mail

user1@user1

E-mail 중복확인

user1@user1은(는) 중복된 이메일입니다. 다른 이메일을 입력해주세요.

*로봇 확인

 로봇이 아닙니다.


reCAPTCHA
개인정보 보호 - 약관

리캡차 확인

localhost:8000 내용:

자동 가입 방지 봇 통과

확인

구현내용

1-1 회원가입

-reCAPTCHA기능

*로봇 확인

✓ 로봇이 아닙니다.

reCAPTCHA
개인정보 보호 · 약관

리캡차 확인

- reCAPTCHA코드

-reCAPTCHA기능

```
<script src='https://www.google.com/recaptcha/api.js'></script>

<script>
$(document)
  .ready(function() {
    $("#test_btn")
      .click(function() {
        $.ajax({
          url : '/webtest/VerifyRecaptcha',
          type : 'post',
          data : {
            recaptcha : $(
              "#g-recaptcha-response")
              .val()
          },
          success : function(
            data) {
            var result = data
              .getElementsByTagName('Integer')[0].childNodes[0].nodeValue;
            switch (parseInt(result)) {
              case 0:
                alert("자동 가입 방지 보 통과.");
                check = 1;
                break;
              case 1:
                alert("자동 가입 방지 보를 통과 하기 위해 꼭 오류가 발생 했습니다. [Error bot Code : "
                  + Number(data)
                  + " ]");
                break;
              default:
                alert("자동 가입 방지 보를 실행 하려 할 오류가 발생 했습니다. [Error bot Code : "
                  + Number(data)
                  + " ]");
                break;
            }
          }
        })
      })
    })
  })
</script>
```

localhost:8000 내용:

자동 가입 방지 봇 통과

확인

구현내용

1-2 로그인 기능

- 카카오 계정으로 로그인
- 네이버 계정으로 로그인
- 아이디 찾기
- 비밀번호 찾기

How About Potato

아이디

비밀번호

로그인

카카오계정으로 로그인

N 네이버로 로그인

아이디 찾기

비번 찾기

회원가입

아이디 찾기

Name:

일용자

E-mail:

user1@user1

확인

취소

당신의 아이디는 user1 입니다.;

다시 시도

닫기

비밀번호찾기

이름:

일용자

아이디:

user1

E-mail:

user1@user1

확인

취소

당신의 비밀번호는 1234 입니다.;

다시 시도

닫기

다중파일로 저장된 상세 캠핑장 이미지를 확인할 수 있고,
객실의 정보를 확인할 수 있다.

구현내용

1-3 회원가입 기능

- 일반회원이 카카오 혹은 네이버로 회원가입

How About Potato

로그인

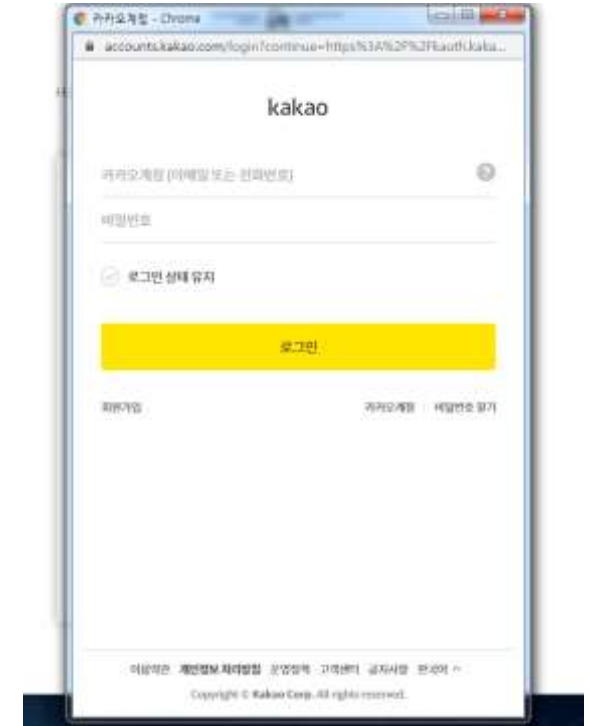
카카오계정으로 로그인

N 네이버로 로그인

아이디 찾기

비번 찾기

회원가입



카카오계정으로 로그인 및 네이버로 로그인 클릭시 각각의 서비스로 회원가입이 가능하다.

구현내용

1-4 회원가입 등록 - 네이버 기능 (로그인시 자동 회원가입)

```
public class NaverLoginAPI {
    private final static String CLIENT_ID = "j1ix933vSAI6kz0Jali";
    private final static String CLIENT_SECRET = "qpcvqz101";
    private final static String REDIRECT_URI = "http://localhost:8000/webtest/wesber_hap/callback";
    private final static String SESSION_STATE = "oauth_state";
    private final static String PROFILE_API_URL = "https://openapi.naver.com/v1/nid/me";

    public String getAuthorizationUrl(HttpSession session) {
        String state = generateRandomString();
        session.setAttribute(SESSION_STATE, state);
        OAuth2Service oauthService = new ServiceBuilder().apiKey(CLIENT_ID).apiSecret(CLIENT_SECRET)
            .callback(REDIRECT_URI).state(state)
            .build(NaverLoginAPI.instance());
        return oauthService.getAuthorizationUrl();
    }
}
```

네이버 세션 통신 함수

```
public OAuth2AccessToken getAccessToken(HttpSession session, String code, String state) throws IOException {
    String sessionState = getSession(session);
    if (StringUtils.pathEquals(sessionState, state)) {
        OAuth2Service oauthService = new ServiceBuilder().apiKey(CLIENT_ID).apiSecret(CLIENT_SECRET)
            .callback(REDIRECT_URI).state(state).build(NaverLoginAPI.instance());
        OAuth2AccessToken accessToken = oauthService.getAccessToken(code);
        return accessToken;
    }
    return null;
}
```

토큰값 저장 함수

```
private String generateRandomString() {
    return UUID.randomUUID().toString();
}

private void setSession(HttpSession session, String state) {
    session.setAttribute(SESSION_STATE, state);
}

private String getSession(HttpSession session) {
    return (String) session.getAttribute(SESSION_STATE);
}

public String getUserProfile(OAuth2AccessToken oauthToken) throws IOException {
    OAuth2Service oauthService = new ServiceBuilder().apiKey(CLIENT_ID).apiSecret(CLIENT_SECRET)
        .callback(REDIRECT_URI).build(NaverLoginAPI.instance());
    OAuthRequest request = new OAuthRequest(Verb.GET, PROFILE_API_URL, oauthService);
    oauthService.signRequest(oauthToken, request);
    Response response = request.send();
    return response.getBody();
}
```

세션 데이터 저장 / 가져오기 / 프로필 API 호출

```
public class NaverLoginAPI extends DefaultApi20 {
    protected NaverLoginAPI() {
    }

    private static class InstanceHolder {
        private static final NaverLoginAPI INSTANCE = new NaverLoginAPI();
    }

    public static NaverLoginAPI instance() {
        return InstanceHolder.INSTANCE;
    }

    @Override
    public String getAccessTokenEndpoint() {
        return "https://nid.naver.com/oauth2.0/token?grant_type=authorization_code";
    }

    @Override
    protected String getAuthorizationBaseUrl() {
        return "https://nid.naver.com/oauth2.0/authorize";
    }
}
```

데이터 통신을 위한 DTO이다.

구현내용

1-4 회원가입 등록

- 네이버 기능 (로그인시 자동 회원가입)
- 파일 업로드

버튼 클릭시 동작하는 함수이다.

```
<div class="login-btn">
  <div id="naverIdLogin"></div>
  <div id="naver_id_login">
    <button onclick="location.href='${url }'" type="button"
      class="btn-naver">
      N
      네이버로 로그인
    </button>
  </div>
</div>
```

```
<script type="text/javascript"
  src="https://static.nid.naver.com/js/naveridlogin_js_sdk_2.0.0.js"
  charset="utf-8"></script>
<script type="text/javascript">
  var naverLogin = new naver.LoginWithNaverId(
  {
    clientId: "8XMDz85NBcpraqKT0WuU",
    callbackUrl: "http://localhost:8000/webtest/member_hap/callback",
    isPopup: false,
    loginButton: {color: "green", type: 3, height: 60}
  }
  );
  naverLogin.init();
</script>
<style>
```

```
<div class="container">
  <br>
  <form class="form-inline" action="#" method="post">
    <div class="form-group">
      <h3>네이버 아이디로 회원가입 완료</h3>
      <p>아이디는 네이버 닉네임으로 적용됩니다.</p>
      <button type="button" onclick="location.href=../">메인 페이지 가기</button>
    </div>
  </form>
</div>
```

동작 성공 시 뜨는 페이지이다.

```
@RequestMapping(value = "/member_hap/login", method = { RequestMethod.GET })
public String login(HttpServletRequest request, HttpSession session, Model model) {
  String c_member_id = "";
  String c_member_id_val = "";

  Cookie[] cookies = request.getCookies();
  Cookie cookie = null;

  if (cookies != null) {
    for (int i = 0; i < cookies.length; i++) {
      cookie = cookies[i];

      if (cookie.getName().equals("c_member_id")) {
        c_member_id = cookie.getValue();
      } else if (cookie.getName().equals("c_member_id_val")) {
        c_member_id_val = cookie.getValue();
      }
    }
  }

  String naverAuthUrl = naverLogin80.getAuthorizationUrl(session);
  System.out.println("로그인 URL : " + naverAuthUrl);
  // 로그인
  model.addAttribute("url", naverAuthUrl);

  request.setAttribute("c_member_id", c_member_id);
  request.setAttribute("c_member_id_val", c_member_id_val);

  return "/member_hap/login";
}
```

```
@RequestMapping(value = "/member_hap/callback", method = { RequestMethod.GET, RequestMethod.POST })
public String callback(Model model, @RequestParam String code, @RequestParam String state, HttpSession session,
  HttpServletRequest request) throws IOException, ParseException {
  OAuth2AccessToken authToken = naverLogin80.getAccessToken(session, code, state);

  apiResult = naverLogin80.getUserProfile(authToken); // String JSON jsonResult
  JSONParser parser = new JSONParser();
  Object obj = parser.parse(apiResult);
  JSONObject jsonObj = (JSONObject) obj;
  JSONObject response_obj = (JSONObject) jsonObj.get("response");
  String nickname = (String) response_obj.get("nickname");
  System.out.println(nickname);

  if (apiResult != null) {
    Member_hapDTO dto = new Member_hapDTO();

    JSONObject create1 = response_obj;

    Member_hapDTO flag = mapper.member_read(nickname);

    if (flag != null) {
      request.setAttribute("str", "이미 회원가입이 완료되었습니다.");
      return "/member_hap/preProc";
    } else {
      Object member_email = create1.get("email");
      Object member_birth = create1.get("birthday");
      Object member_name = create1.get("name");
    }
  }
}
```

json 객체로 네이버에서 정보를 받아와 추출후 저장해주는 코드이다.

구현내용

1-5 회원가입 등록

- 카카오키능 (로그인 시 회원가입 창 이동)
- 파일 업로드

```
<div class="login-btn">
  <div id="kakao_id_login">
    <button id="custom-login-btn" type="button" class="btn-kakao"
      onclick="loginWithKakao()">
      <i class="fas fa-comment"></i>
      카카오계정으로 로그인
    </button>
  </div>
</div>
```

버튼 클릭시 동작하는 함수이다.

```
Kakao.init('40c7acb685fb1a07ba5cb37a10f4029e');
function loginWithKakao() {
  Kakao.Auth.login({
    success: function(authObj) {
      Kakao.API.request({
        url: '/v1/user/me',
        success: function(res) {
          console.log(JSON.stringify(res.kaccount_email));
          console.log(JSON.stringify(res.id));
          console.log(JSON.stringify(res.properties.profile_image));
          console.log(JSON.stringify(res.properties.nickname));
          function sendPost(action, params) {
            var form = document.createElement('form');
            form.setAttribute('method', 'post');
            form.setAttribute('action', action);
            document.charset = "utf-8";
            for ( var key in params) {
              var hiddenField = document.createElement('input');
              hiddenField.setAttribute('type', 'hidden');
              hiddenField.setAttribute('name', key);
              hiddenField.setAttribute('value', params[key]);
              form.appendChild(hiddenField);
            }
            document.body.appendChild(form);
            form.submit();
          }
          sendPost('member_kakao_create', {
            'name': res.properties.nickname,
            'email': res.kaccount_email,
            'id': res.id,
            'fname': res.properties.profile_image
          });
        }
      });
    },
    fail: function(err) {
      alert(JSON.stringify(err));
    }
  });
}
```

kakao 함수 호출 시 동작 코드

kakao와 통신하는 토큰 값

```
@RequestMapping("/member_hap/member_create")
public String member_create(HttpServletRequest request, Model model) {
  if (request.getParameter("id") != null) {
    String kakao_id = request.getParameter("id");
    String kakao_email = request.getParameter("email");
    String kakao_fname = request.getParameter("fname");
    String kakao_name = request.getParameter("name");

    model.addAttribute("kakao_id", kakao_id);
    model.addAttribute("kakao_email", kakao_email);
    model.addAttribute("kakao_fname", kakao_fname);
    model.addAttribute("kakao_name", kakao_name);

    return "/member_hap/member_create";
  } else {
    return "/member_hap/member_create";
  }
}
```

kakao 정보를 받아온다.

```
if (request.getParameter("set").equals("Kakao")) {
  int sizeF = (int) dto.getMember_backfileMF().getSize();

  dto.setMember_fname(request.getParameter("member_fname"));
  if (sizeF > 0) {
    dto.setMember_backfile(utility.saveFileSpring(dto.getMember_backfileMF(), basePathF));
  } else {
    dto.setMember_backfile("back.jpg");
  }
  int flag = mapper.member_create(dto);

  if (flag == 1) {
    return "redirect:/";
  } else {
    model.addAttribute("str", "알 수 없는 이유로 회원가입에 실패하였습니다.");
  }
}
```

view에서 kakao와 일반 회원을 구분 지어서 판별

구현내용

1-6 메뉴 구분 바

- 비회원 메인 화면 상태



- 관리자 로그인



- 일반 회원 로그인



구현내용

2-1 공지사항

- 주요 공지사항은 상단에 고정기능
- 삭제 수정은 작성자와 관리자만

번호	제목	작성자	날짜	조회수
17	123	관리자	2019-12-20	1
8	개항물다	관리자	2019-12-20	7
7	공지사항 상단1	관리자	2019-12-20	2
6	공지사항 상단2	관리자	2019-12-18	3
1	공지사항 상단	관리자	2019-12-18	12
16	123	관리자	2019-12-20	174
15	123	관리자	2019-12-20	3
14	123	관리자	2019-12-20	1
13	123	관리자	2019-12-20	0
12	1	관리자	2019-12-20	20
11	1	관리자	2019-12-20	0
10	1243	관리자	2019-12-20	17
9	123	관리자	2019-12-20	0
5	공지사항 테스트	관리자	2019-12-18	8
4	공지사항 테스트	관리자	2019-12-18	48

1 2



관리자 계정만 보이는 화면

- 회원 / 비회원의 경우 등록 수정 삭제 불가능

공지사항 게시판 화면

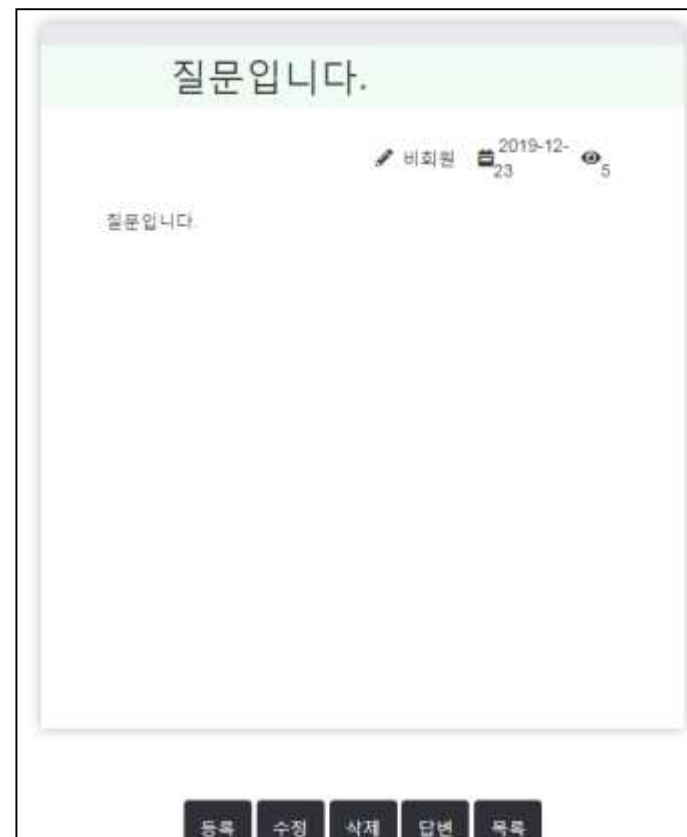
구현내용

2-2 QNA

- 주요 공지사항은 상단에 고정기능
- 삭제 수정은 작성자와 관리자만

번호	제목	조회수	작성자
6	공지사항입니다.	1	관리자
4	QNA 작성 요령	3	관리자
3	QNA 게시판입니다	3	관리자
5	질문입니다.	3	비회원
2	테스트	1	user1
1	테스트	2	user1

질문 / 답변 게시판 화면



관리자 계정만 보이는 화면

- 회원 / 비회원의 경우 등록 수정 삭제만 가능

구현내용

3-1 숙소 목록

- 주소로 검색 기능 (클릭 시 원하는 곳 선택)
- (관리자)업소정보 수정, 업소 삭제기능
- 상단에 고정숙소 (유료 결제 시)



```
<form class="form-inline" method="post" action="list">
  <div class="form-group">
    </div>
  <br>
  <div class="list-mini-list">
    <div style="cursor: pointer;" class="list-mini-container"
      onclick="javascript:list('')">
      
      <div class="list-mini-container">전국</div>
    </div>

    <div style="cursor: pointer;" class="list-mini-container"
      onclick="javascript:list('강원')">
      
      <div class="list-mini-container">강원</div>
    </div>
  </div>
```


구현내용

3-2 숙소 등록

- 숙소 이름 중복체크
- 카카오 우편번호 api사용 db에 저장
- 상단에 고정숙소

숙소 등록

(* 필수입력사항)

업소 소개 사진

파일 선택

선택된 파일 없음

*업소 이름

test호텔

중복체크

test호텔은(는) 사용 가능한 숙소이름 입니다.

우편번호:

우편번호

주소검색

주소:

주소

상세주소

계좌:

입금받을 계좌번호를 입력하세요

숙소안내사항:

숙소 안내사항을 입력하세요

등록

취소

예) 판교역로 235, 분당 주공, 삼성동 681

tip

아래와 같은 조합으로 검색을 하시면 더욱 정확한 결과가 검색됩니다.

도로명 + 건물번호

예) 판교역로 235, 제주 첨단로 242

지역명(동/리) + 번지

예) 삼성동 681, 제주 영평동 2181

지역명(동/리) + 건물명(아파트명)

예) 분당 주공, 연수동 주공3차

사서함명 + 번호

예) 분당우체국사서함 1~100

```
<script src="http://dmaps.daum.net/map_js_init/postcode.v2.js"></script>
<script>
function sample6_execDaumPostcode() {
    new daum.Postcode({
        oncomplete : function(data) {
            var addr = ''; // 주소 변수
            var extraAddr = ''; // 참고항목 변수

            if (data.userSelectedType === 'R') { // 사용자가 도로명 주소를 선택했을 경우
                addr = data.roadAddress;
            } else { // 사용자가 지번 주소를 선택했을 경우(J)
                addr = data.jibunAddress;
            }

            if (data.userSelectedType === 'R') {
                if (data.bname !== '' && /[읍|로|가]$/g.test(data.bname)) {
                    extraAddr += data.bname;
                }
                if (data.buildingName !== '' && data.apartment === 'Y') {
                    extraAddr += (extraAddr !== '' ? ', ' : '')
                        + data.buildingName : data.buildingName);
                }
                if (extraAddr !== '') {
                    extraAddr = ' (' + extraAddr + ')';
                }
            } else {
            }
            document.getElementById('board_zipcode').value = data.zonecode;
            document.getElementById("board_address1").value = addr;
            document.getElementById("board_address2").focus();
        }
    }).open();
}
</script>
```

구현내용

3-3 숙소 상세 정보



-방 사진들 슬라이드 쇼

방 사진	방 이름	방 가격	침대 숙박인원	방 정보
	일반실	60000원	2명	일반실 정보
	특실	80000원	4명	특실 정보
	파티룸	100000원	8명	파티룸 정보

-관리자 계정 권한 방등록, 정보수정, 삭제

수정/삭제

방 등록

방정보수정

방사진수정

방삭제

구현내용

3-3 숙소 상세 정보 - 위치



지도에 위치와 숙소이름을 표시

-숙소 등록시 입력한 주소를 카카오 API를 이용해 검색,

-관리자 계정 권한 방등록, 정보수정, 삭제

수정/삭제

방정보수정

방사진수정

방삭제

```
<script type="text/javascript"
src="//dapi.kakao.com/v2/maps/sdk.js?appkey=a3cb1a887ea201052c452feb4c0f8ed0&libraries=services,clusterer,drawing"></script>
<div class="read-map">
<h1>숙소 지도</h1>
<p>주소 : ${dto.board_address1 } ${dto.board_address2 }
<div class="read-map" id="map" style="width: 100%; height: 350px;">
</div>
<script>
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
mapOption = {
center : new kakao.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
level : 3
};

var map = new kakao.maps.Map(mapContainer, mapOption);
var geocoder = new kakao.maps.services.Geocoder();

geocoder
.addressSearch(
'${dto.board_address1}',
function(result, status) {

if (status === kakao.maps.services.Status.OK) {
var coords = new kakao.maps.LatLng(
result[0].y, result[0].x);

var marker = new kakao.maps.Marker({
map : map,
position : coords
});

var infowindow = new kakao.maps.InfoWindow({
content : '<div style="width:150px;text-align:center;padding:6px 0;">${dto.board_name}</div>'
});
infowindow.open(map, marker);
map.setCenter(coords);

}
});
</script>
</div>
```

숙소 위치 상세 보기 코드

구현내용

3-3 숙소 상세 정보 - 리뷰

리뷰 등록

내용

별점 (아래를 클릭해주세요 !!)

★★★★☆

등록 취소

리뷰 등록하는 화면

리뷰

user1

별점: 5.0 (5점 만점)

별점: ★★★★★

리뷰가 등록된 화면

```
function() {
  var board_num = "${board_num}";
  var sno = "${sno}";
  var eno = "${eno}";
  var replyUl = $(".chat");

  showList();

  function showList() {
    replyService
      .getList(
        {
          board_num : board_num,
          sno : sno,
          eno : eno
        },
        function(list) {
          var str = "";

          if (list == null || list.length == 0) {
            return;
          }

          for (var i = 0, len = list.length; i < len; i++) {
            str += "<li class='list-group-item' data-rv_num='" + list[i].rv_num + "'>";
            if (list[i].member_frame
              .includes("http")) {
              str += "<div><div class='header'><div><img src='" + list[i].member_frame + "' alt='\" + list[i].member_id + "\" height='100px' width='100px'></div><strong class='primary-font'>{nicknames}</strong></div></div>";
            } else {
              str += "<div><div class='header'><div><img src='" + list[i].member_frame + "' alt='\" + list[i].member_id + "\" height='100px' width='100px'></div><strong class='primary-font'>{nicknames}</strong></div></div>";
            }

            str += "<small class='pull-right text-muted'>";
            str += list[i].rv_date;
            str += "</small></div><div>{content}</div>";
            str += "</div></div>";

            str += "<div><p><strong>{rv_star}</strong></p></div></li>";
            str += list[i].rv_star;
            str += "</div></div>";
          }

          replyUl.html(str);
          showOnlyPage(1);
        }
      );
  }
}
```

리뷰 작성 관련 코드

구현내용

3-3 숙소 상세 정보 – 숙소 정보



숙소 정보 화면 로그인 시



숙소 정보 화면 비로그인 시

```
<div class="read-submit">
  <c:choose>
    <c:when test="${not empty sessionScope.member_id}">
      <button type="submit" class="btn"
        onclick="location.href='../res/create?board_num=${dto.board_num}'">예약하기</button>
    </c:when>
    <c:otherwise>
      <button type="submit" class="btn"
        onclick="location.href='../member_hap/login'">로그인</button>
    </c:otherwise>
  </c:choose>
</div>
```

구분 코드

구현내용

4-1예약 및 결제

- 마이페이지에서 예약취소 가능

예약정보입력

성명(예약자)

일용자

전화번호

010-0000-0000

E-MAIL

user1@user1

성명(실사용자) *

※ 회원명으로 예약되며, 모든 연락은 실사용자로 진행됩니다.

휴대전화(실사용자) *

남기실말

입실일*

연도-월-일

퇴실일*

연도-월-일

층인원*

2

※ 기준인원 초과 시 인원추가비용 발생

※ 영유아 or 방문객 포함 최대인원 초과 시 예약 불가합니다.

결제방식

무통장입금

※ 반드시 고객센터에서 안내 받으신 후 진행 해주십시오.

※ 결제금액은 우선안내드립니다.

입금계좌정보

기업은행 110-397-167909

입금자명 *

현금영수증 123

☐ 등록된 핸드폰 번호로 처리되며, 변경 시 해당 업소에 연락하시길 바랍니다.

상담 접수

다시 작성

예약 등록하는 화면

약관 내역

팬션/콘도/카리반 예약 주의사항에 동의 내용

1. 홈페이지를 통하여 예약 진행시 회원 혜택을 받을 수 있습니다.(현장 직접 예약시 혜택 불가)
2. 당일 예약은 불가하며, 토/일/월요일 이튿날 금요일 오후 16시까지 입금확인 시 예약이 완료됩니다.
3. 객실은 입금 선착순 마감됩니다.(입금 안내 문자 전송 후 24시간 이내 미입금 시 예약 자동취소)
4. 기준인원에 대한 객실 요금입니다.(숙박업체 규정 안내사항)
(기준인원 초과 시 1인당 인원추가 비용 발생 / 최대 인원 초과 시 예약 완료라도 입실이 거부 될 수 있으며, 예약 금액은 반환되지 않습니다.)
5. 예약 완료 후 취소 시에는 해당 숙박업체의 환불 규정에 따라 취소금이 발생합니다.(예약완료 후 변경 시에도 동일 적용)
6. 고객센터 이용 시간(10시 - 17시) 이외의 요청은 익일로 처리되오니 참고 바랍니다.
7. 해당 숙박 업체별 입/퇴실 시간을 준수해야 합니다.
8. 미성년자는 예약을 받지 않습니다.
9. 애완동물과 함께 입실 할 수 없습니다.
10. 제휴처는 당사의 사정에 따라 예고 없이 변경 될 수 있습니다.
11. 설수기 / 연휴 기간에는 이용 금액과 환불 규정이 변경 될 수 있습니다.
12. 패밀리와 픽서리 이틀은 총 월 2회까지 가능합니다.(연박가능 / 1일 2객실 사용불가)
13. 당사는 통신판매 증가자로 팬션운영은 팬션주의 책임하에 운영되고 있습니다. 팬션 현장에서 일어난 일에 대해서는 당사의 책임이 없습니다.
14. 당사는 숙박상품의 판매대행처로 소득세법과 부가가치세법에 따라 세금계산서 발행이 불가합니다.(단, 현금영수증 발급은 가능)

이동식 규의사항에 동의 내용

보유 금액은 기준인원에 대한 객실 요금입니다. 기준인원 초과 시 1인당 인원추가 비용이 발생합니다.
 사전에 협의 되지 않은 방문객 또는 열·유아 포함 최대인원 초과 시에는 예약완료임에도 불구하고 입실이 거부 될 수 있으며, 예약 금액은 반환되지 않습니다.
 ※ 인원추가비용: 0세 부터 4만원
 ※ 입실시간: 오후 3시 , 퇴실시간: 오전 11시
 (22시 이후의 입실은 사전에 반드시 연락을 주시기 바랍니다.)
 ※ 부대시설 : 바베큐
 - 바베큐 (숯,그릴) 2인 15,000원
 ※ 환불 규정 안내 (이유일 기준)
 기본 취약금 10%, 5일전 90% 환불, 4일전 80% 환불, 3일전 50% 환불, 1일전 30% 환불, 당일 환불금액 없음
 미용차가 마성년자가 아니며 성인임에 동의 내용

☐ 모두 동의합니다

예약 기본 약관 화면

예약 하기

선택하기	객실명	기준인원	최대인원
<input type="radio"/> 선택	일반실	2명	2명
<input type="radio"/> 선택	특실	2명	4명
<input checked="" type="radio"/> 선택	파티룸	2명	8명

방 선택 화면

구현내용

4-2 예약 확인

- 마이페이지에서 예약취소 가능

예약 확인

예약 시 받은 예약 번호를 정확하게 입력하세요

검색

비 로그인 시
예약 번호로 조회

예약 내역	
예약번호	67682270
아이디	user1
성명	김용자
이메일	user1@user1
실사용자(성명)	이재준
실사용자 연락처 [휴대전화]	010-9995-0547
보건 내용	14시에 체크인 합니다
휴대전화번호	010-0000-0000
예약접수날짜	2020-01-13
숙박 상호	test호텔
숙박명	일반실
입실일시	2020-01-13
퇴실일시	2020-01-14
결제 내역	
입금계좌	기일은행 110-397-167909
입금자명	이재준
입금액	90000원
현금영수증	010-9995-0547

예약 상세 정보 화면

구현내용

4-3 예약 목록(관리자기능)

- 마이페이지에서 예약취소 가능

예약 관리 화면

예약 번호	예약 이름	예약 날짜	승인 여부	승인 처리	취소 처리
67682270	이재근	2020-01-13	예약 요청	[승인 처리]	[취소 처리]
62664276	이재근	2020-01-04	예약 요청	[승인 처리]	[취소 처리]

예약내역

예약번호 : 67682270

예약자명 : 이재근

사용인원 : 2

입실일시 : 2020-01-13

퇴실입실 : 2020-01-14

[예약정보 상세보기]

예약취소

예약 취소 요청

예약수정

예약 수정하기

예약상태

예약 요청 중

예약번호 : 62664276

예약자명 : 이재근

사용인원 : 2

입실일시 : 2020-01-04

퇴실입실 : 2020-01-05

[예약정보 상세보기]

예약취소

예약 취소 요청

예약수정

예약 수정하기

예약상태

예약 요청 중

예약 승인 / 취소 화면

↓

예약 승인 처리

성명

이재근

연락처

010-9995-0547

인원

2

숙박 상호

test호텔

숙박명

일반실

입실일시

2020-01-13

퇴실일시

2020-01-14

승인여부

예약 요청

승인완료

승인완료

보류상태

수정완료

거절완료

처리

목록 가기

구현내용 5-1 마이 페이지

- 마이페이지에서 예약취소 가능



비밀번호 변경

비밀번호

비밀번호 확인

별명 변경

기존 별명:

별명:

회원수정

로그인/회원관리

이름: 홍길동

성별: 남성

생년월일: 1990-01-01

전화번호: 010-0000-0000

이메일: user1@user1

수정 삭제

로그인 확인

[illegible]

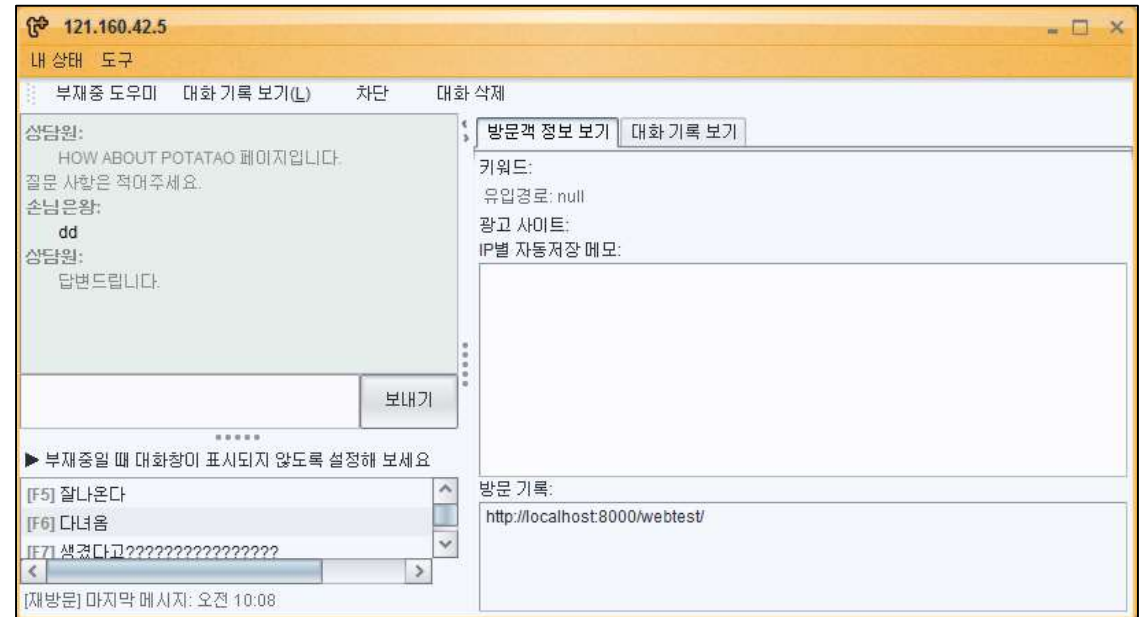
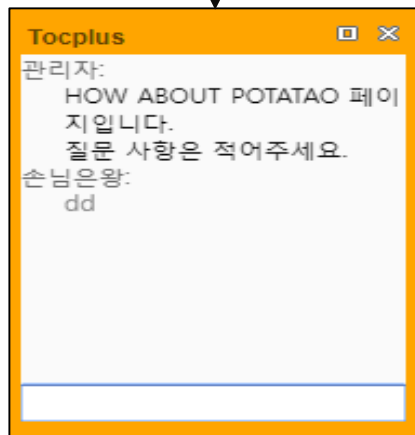
구현내용

6-1 상담 챗봇

- 상담사와 실시간 채팅 기능

```
<script type="text/javascript">
  tocplusTop = 1150;
  tocplusLeft = 5;
  tocplusMinimizedImage = 'http://kr03.tocplus007.com/img/minimized_ko.gif';
  tocplusHAlign = 'right';
  tocplusLocaleFixed = 'ko';
  tocplusWidth = 180;
  tocplusHeight = 220;
  tocplusUserName = '손님은왕';
  tocplusFrameColor = '#FFA500';
  tocplusFloatingWindow = true;
  var tocplusHost = (("https:" == document.location.protocol) ? "https://"
    : "http://");
  document
    .write(unescape("%"
    + "3Cscript src='"
    + tocplusHost
    + "kr03.tocplus007.com/chatLoader.do?userId=jorney0' type='text/javascript'"
    + "%" + "3E" + "%" + "3C/script" + "%" + "3E"));
</script>
```

talk plus 챗봇 코드



챗봇 관리자 화면

구현내용 7-1 기타

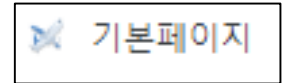
- 파비콘 기능
- 외부 도메인 연결

```
<link rel="shortcut icon" href="${pageContext.request.contextPath }/images/myicon.ico" type="image/x-icon">
<link rel="icon" href="${pageContext.request.contextPath }/images/myicon.ico" type="image/x-icon">
```

파비콘 설정코드



설정할 아이콘



설정된 아이콘

전체 1건

바로 가기

도메인 또는 일련번호(IP) 입력

확장 검색

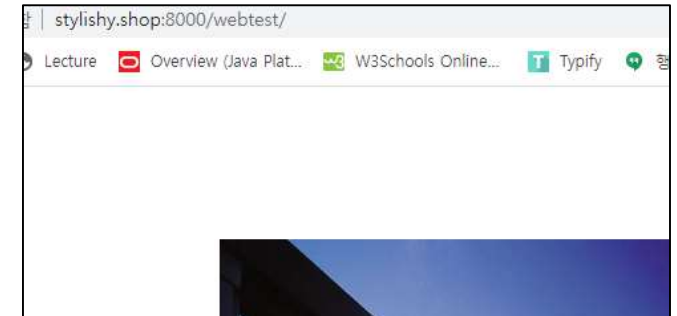
10개

서비스	도메인/일련번호(IP)	시작일	만기일	기준 가격	바로 가기
도메인	stylishy.shop	2019-12-20	2020-12-20	53,900원/년	관리중

외부 도메인 구입

DNS 설정						
stylishy.shop						
레코드 개수: 2개 · 최근 업데이트: 2019-12-23 09:37:19 · 네임서버: ns.gabia.co.kr						
타입	호스트	값	TTL	우선 순위	리버스	상태
A	@	172.16.01.188	3600		CNS 설정	수정 삭제
A	www	172.16.01.188	3600		CNS 설정	수정 삭제

외부 도메인 연결



설정된 모습

감사합니다