

i부터 시작하는 코딩 인강



TEAM 3 : 유승관 : 김재동 : 안예인 : 차승혁 : 김정진

CONTENTS

01

프로젝트 개요

- 1-1. 개발 동기
- 1-2. 기획 의도
- 1-3. 팀원 소개

02

진행 프로세스

- 2-1. 개발 일정
- 2-2. 개발 환경

03

프로젝트 상세

- 3-1. 구조도
- 3-2. ERD
- 3-3. 기술 설명

04

프로젝트 시연

- 4-1. 후기

01 프로젝트 개요

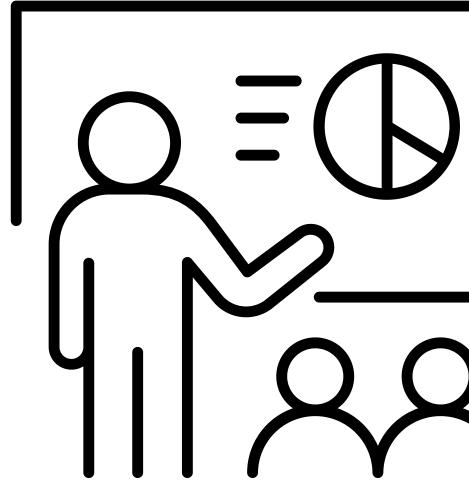
1-1. 개발 동기

1-2. 기획 의도

1-3. 팀원 소개

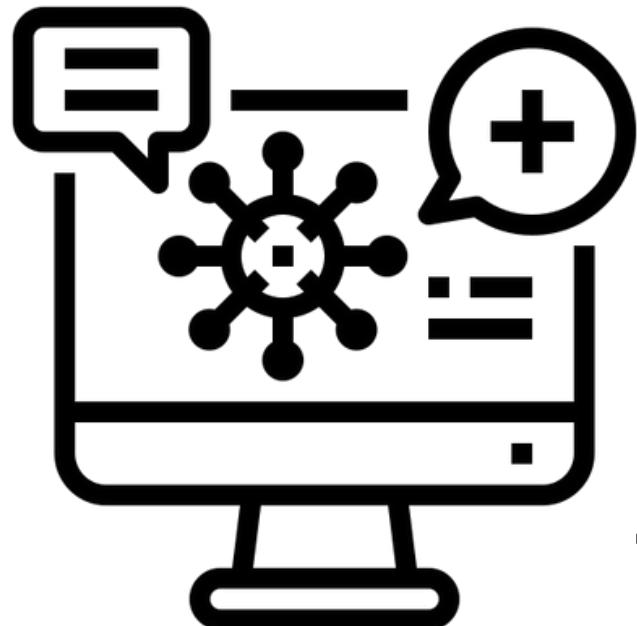
01

프로젝트 개요 1-1. 개발 동기



코딩 전국 학원수

722곳 이상



코딩 전문 인터넷강의 사이트

대표 5개이만



코딩관련 유튜브 조회수

최대 260만뷰

어디서부터
시작해야
하는 걸까?!



01

프로젝트 개요

1-1. 개발 동기

전체 강의

전체 강의 검색 검색

기술 검색 Spring Java Spring Boot Node.js Python Kotlin Express Docker Blender

온라인 오프라인 무료 유료 할인중 New 입문 초급 중급이상

추천순



Frontend Course
HTML CSS JS

[인프런x코드캠프] 부트캠프에서 만든 고농축 프론트엔드 코스

코드캠프

★★★★★ (38) ₩396,000 ₩304,920

+1,000명 할인중



Spring MVC Part.1
Back-End Web Dev. Basic Course

스프링 MVC 1편 - 백엔드 웹 개발 핵심 기술

김영한

★★★★★ (1539) ₩99,000 ₩76,230

+18,300명 할인중



Java to Kotlin Starter Guide

자바 개발자를 위한 코틀린 입문 (Java to Kotlin Starter Guide)

최태현

★★★★★ (106) ₩55,000 ₩42,350

+1,500명 할인중



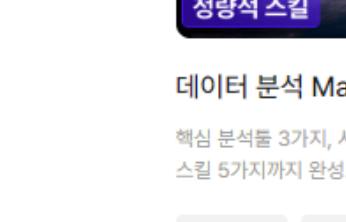
First-Come Event System
Hands-On Learning

실습으로 배우는 선착순 이벤트 시스템

최상용

★★★★★ (10) ₩10,800 ₩15,250

+400명 세강의 할인중



PYTHON BASIC TO ADVANCED
LEARN FROM SILICON VALLEY ENGINEER

실리콘밸리 엔지니어가 가르치는 파이썬 기초부터 고급까지

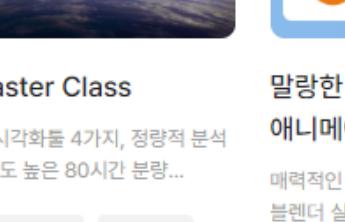
Seungjoon Lee

★★★★★ (10) ₩60,500 ₩46,580

넓고 알게 외워서 컴공 전공자 되기

널널한 개발자

★★★★★ (68) ₩66,000 ₩50,820



Computer Science for beginners

three.js로 시작하는 3D 인터랙티브 웹

1분코딩

★★★★★ (61) ₩165,000 ₩127,050

신규 런칭 강의

모든 신규강의 보기 >



데이터 분석
데이터 시각화
정량적 스킬

데이터 분석 Master Class

핵심 분석 3가지, 시각화 4가지, 정량적 분석 스킬 5가지까지 완성도 높은 80시간 분량...

데이터분석 데이터시각화 머신러닝

블렌더 애니메이션 3D 캐릭터



2023 Dart 수록
업종별 재무제표 분석

말랑한 캐릭터 표현을 위한 11가지 애니메이션 실습

매력적인 캐릭터 애니메이션을 위한 11가지 블렌더 실습! 카카오프렌즈, 당근마켓 등과 협업...

블렌더 애니메이션 3D 캐릭터

업종별 Dart 공시 재무제표



ChatGPT
블로그 수익화
평생 소장

ChatGPT와 함께 월 100만원 수익 블로그 만들기

5시간으로 마스터하는 ChatGPT 활용 수익형 블로그! 코딩 지식과 블로그 운영 경험이 없어도...

ChatGPT 블로그 수익화

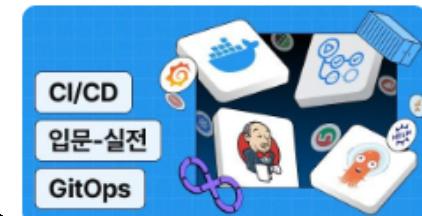


디지털 리터러시

직장인 생존을 위한 디지털 리터러시 by. 김지현

내 직업 수명은 얼마나 남았지? 대한민국 최고 IT 전문가 SK 김지현 부사장만의 인사이트를 통해...

IT실습 비즈니스 문해력 IT



Python Level 1

CI/CD 입문-실전 GitOps

격차 패키지: 한 번에 끝내는 CI/CD의 모든 것: Docker부터...

원계 프로파 셜 캐릭터드로잉



프론트엔드 시나리오
실전 문제 해결 100선

네오아카데미 : 3명의 아트디렉터와 슬럼프 구간 돌파 KIT

더욱 성장하고 싶은 프로 일러레 지망생들을 위해 준비한 슬럼프 구간 돌파 KIT. 네오아카데미...

프론트엔드 실무 주니어 미드레벨



디지털 리터러시

직장인 생존을 위한 디지털 리터러시 by. 김지현

내 직업 수명은 얼마나 남았지? 대한민국 최고 IT 전문가 SK 김지현 부사장만의 인사이트를 통해...

IT실습 비즈니스 문해력 IT



순서없이 나열만 되어있는 강의

어디서, 어떤 강의
를 들어야 할까?



01 프로젝트 개요

1-1. 개발 동기

1-2. 기획 의도

1-3. 팀원 소개

01

프로젝트 개요 1-2. 기획 의도



보기 쉬운 홈페이지



소통하는 강의

01 프로젝트 개요

1-1. 개발 동기

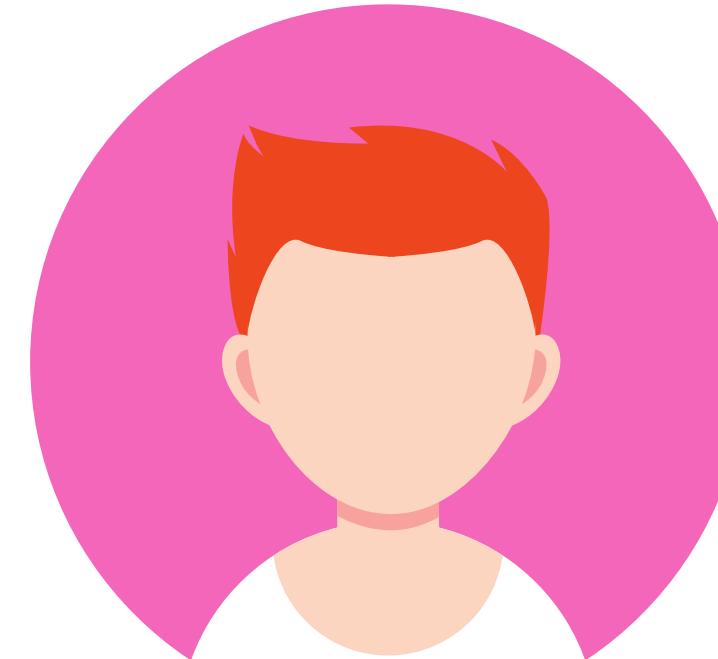
1-2. 기획 의도

1-3. 팀원 소개



유승관

- 프로젝트 총괄
- Spring MVC - 건의 / QnA 게시판
- 언어 카드 & 오시는 길
- 안드로이드 앱 로그인/회원가입



김재동

- BACKEND 기술구현 총괄
- Header(NavBar)
- Spring MVC - 로그인 & 로그아웃
- Spring MVC - 회원가입
- Spring MVC - 공지사항/이벤트 게시판
- IAMPORT - 카드결제



차승혁

- FRONTEND 디자인 총괄
- 강의 영상 & 상세 페이지
- SPRING MVC - 로그인, 회원가입
- SPINGBOOT, REACT - 로그인, 자유 게시판



안예인

- 수강후기 & D-day 카운트 다운
- 팝업창(쿠키적용) 반응형 웹
- REACT - 채널톡
- Spring MVC - 에러해결 게시판



김정진

- 언어 카드 & 오시는 길
- 강사 소개 & Footer
- SPRING MVC - QnA 게시판
- SPRINGBOOT, REACT - 로그인
- 안드로이드 앱 로그인/회원가입

02 진행 프로세스

2-1. 개발 일정

2-2. 개발 환경

02

진행 프로세스

2-1. 개발일정

GIT HUB

The screenshot shows a GitHub repository named 'Team Project'. The repository has 3 commits and 23 pushes. It contains files like 'src', 'target', '.classpath', '.project', '.springBeans', 'OneCoin_ERMaster.erm', and 'pom.xml'. A 'Todos' board is visible on the right side, divided into 'Todo' (3 items), 'In Progress' (29 items), and 'Done' (29 items) columns.

TRELLO

The screenshot shows a Trello board titled '개발 3팀 최종 프로젝트'. The board is organized into several lists corresponding to dates from June 28 to July 6. Each list contains cards representing different tasks or features, such as '프로젝트 시작', '개발 연어 작성(한글)', 'navbar 수정', and 'Footer 수정'. The board is currently in 'Workspace visible' mode.

GANTT CHART

※ 개발 3팀 프로젝트명 =One코인(1부터 시작하는 코딩 조보자들을 위한 1등 코딩 인				완료여부 표시																									
구분	수행 사항	담당자	시작일	종료일	진행 기간	완료 여부	6월			7월																			
							28일	29일	30일	1일	2일	3일	4일	5일	6일	7일	8일	9일	10일	11일	12일	13일	14일	15일	16일	17일	18일	19일	20일
공통	개요 및 layout 작성 완료		6/28	6/28	1	진행완료																							
	프런트 메인 페이지 제작			6/29	7/21	23	진행완료																						
	종류별 게시판 제작			7/5	7/21	17	진행완료																						
	회원가입/로그인			7/4	7/17	14	진행완료																						
	결제창			7/17	7/19	3	진행완료																						
	스프링부트/리액트로그인			7/13	7/17	5	진행완료																						
	안드로이드 앱 회원가입/로그인			7/18	7/21	4	진행완료																						

02 진행 프로세스

2-1. 개발 일정

2-2. 개발 환경



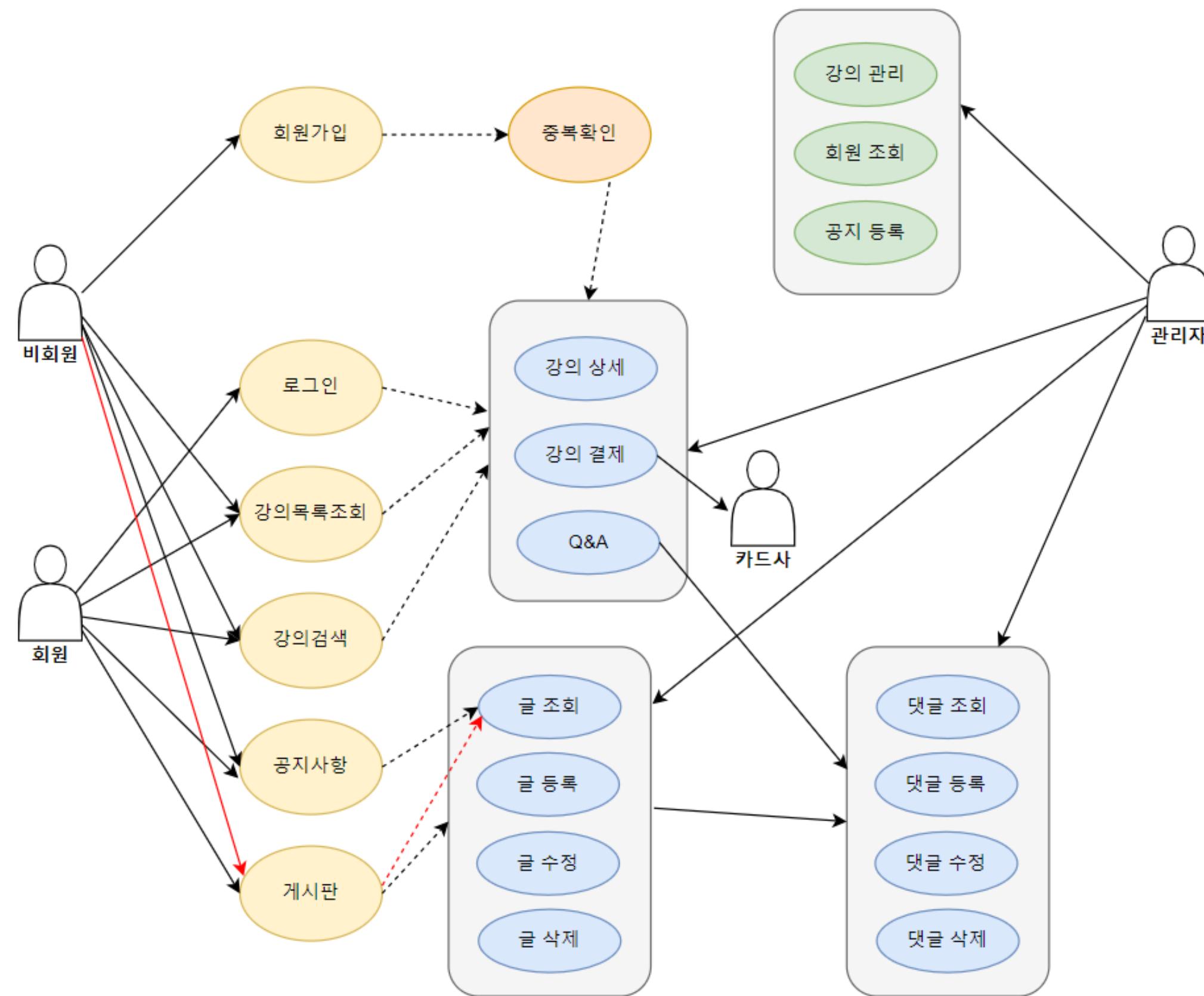
03 프로젝트 상세

3-1. 구조도

3-2. ERD

3-3. 기술 설명

3-1. 구조도(use case diagram)



03 프로젝트 상세

3-1. 구조도

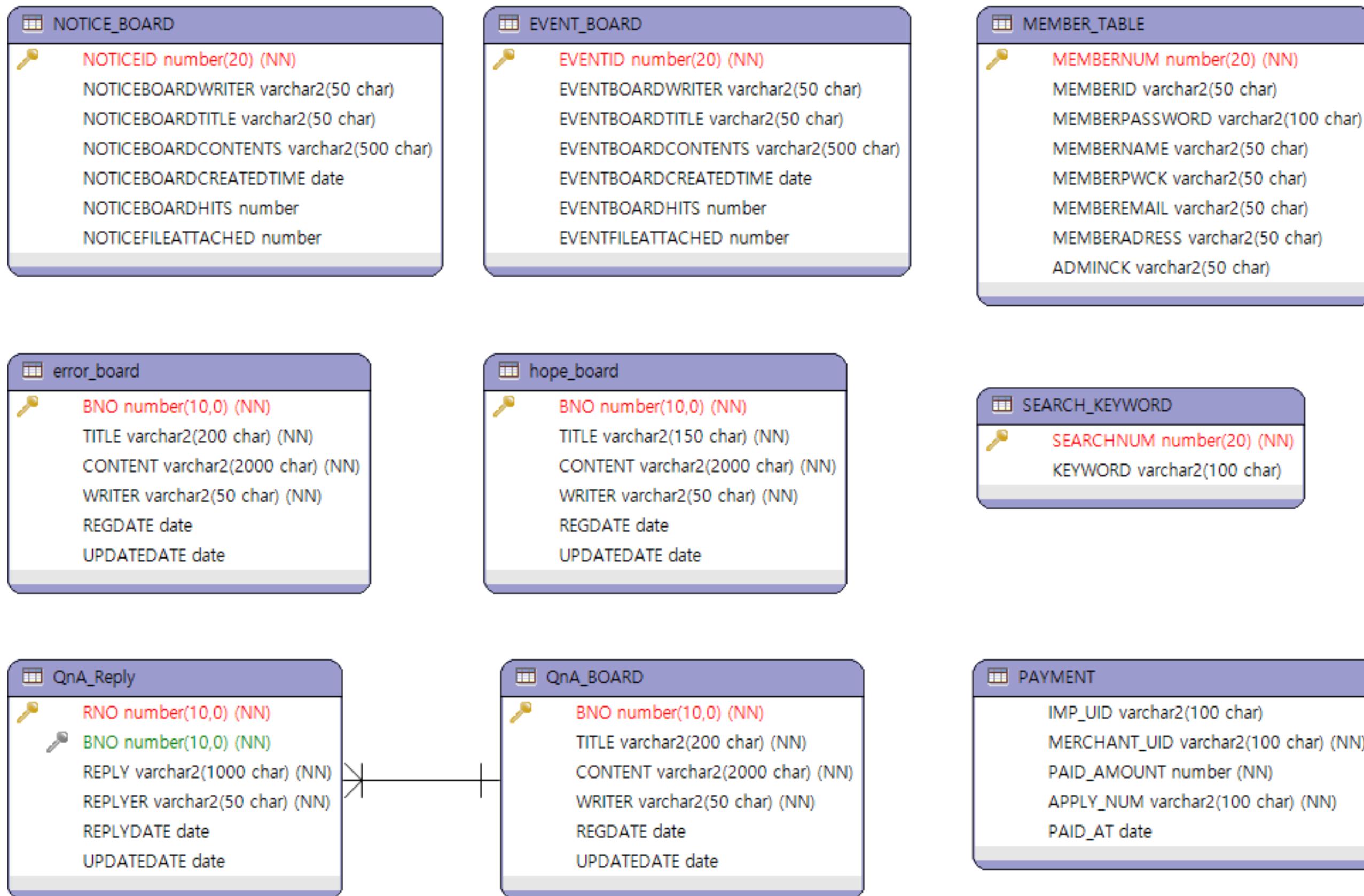
3-2. ERD

3-3. 기술 설명

03

프로젝트 상세

3-2. ERD(Entity Relationship Diagram)



03 프로젝트 상세

3-1. 구조도

3-2. ERD

3-3. 기술 설명

03

프로젝트 상세

3-3. 기술 설명

회원가입

아이디

비밀번호

이름

비밀번호 확인

이메일

인증번호전송

인증번호를 입력해주세요!

주소

주소찾기

상세주소 (필수 아님)

개인정보 수집 및 이용에 동의합니다.

[가입완료](#)

회원가입



Controller

```
@PostMapping("/signup")
public String Signup(@ModelAttribute MemberDTO dto, Model model) {
    int saveResult = memberService.save(dto);
    if (saveResult > 0) {
        return "redirect:/";
    } else {
        System.out.println("오류");
        return "/";
    }
}

<insert id="save" parameterType="member">
    insert into member_table (memberNum, memberId, memberPassword,
    memberName, memberPwCk, memberEmail, memberAdress, adminCk)
    values(member_seq.nextval, #{memberId}, #{memberPassword},
    #{memberName}, #{memberPwCk}, #{memberEmail}, #{memberAdress}, 0)
</insert>
```

XML

아이디 ~ 주소
JS, Jquery로
검증구현,
아이디중복기능 구현
smtp, MimeMessage
등을 이용하여
메일 인증번호전송
및 검증구현
다음 카카오 API를
이용하여 주소찾기 구현

03

프로젝트 상세

3-3. 기술 설명

로그인 & 로그아웃

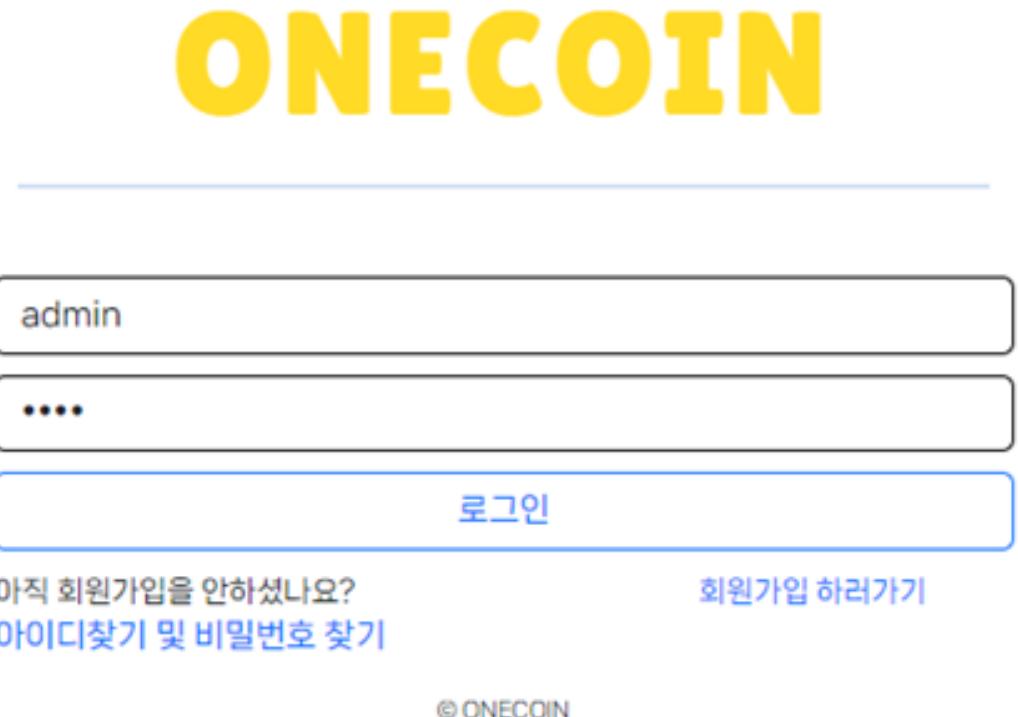
Controller

```
@PostMapping("/login")
public String login(@ModelAttribute MemberDTO dto, Long memberNum,
    Model model,
    HttpServletRequest request, RedirectAttributes rttr) {

    HttpSession session = request.getSession();
    MemberDTO lvo = memberService.login(dto);

    if(lvo == null) {
        int result = 0;
        rttr.addFlashAttribute("result", result);
        return "redirect:/member/login";
    }

    session.setAttribute("member", lvo);
    return "redirect:/";
}
```



© ONECOIN

Controller

```
@GetMapping("/logout.do")
public String logoutMainGET(HttpServletRequest request) {
    HttpSession session = request.getSession();

    session.invalidate();
    return "redirect:/";
}
```

로그인 시 서버에 값을
저장하기 위하여
session 이용하여
로그인 구현

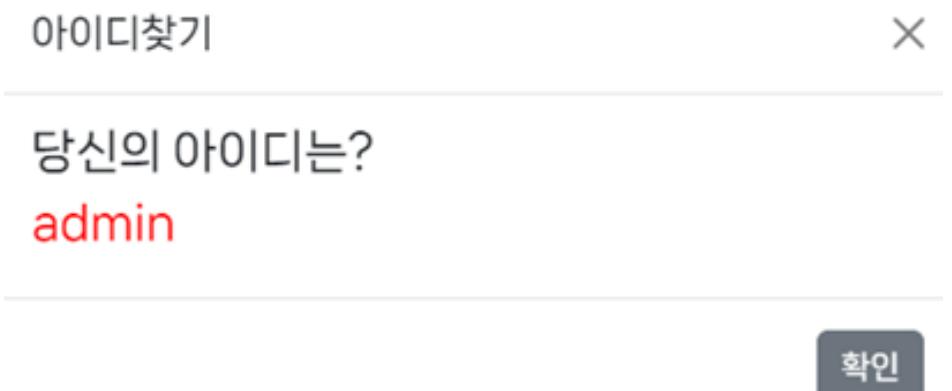
session 값을 지움으로써
로그아웃 구현

03

프로젝트 상세

3-3. 기술 설명

아이디 찾기



Controller

```
@PostMapping("/userSearch")
@ResponseBody
public String search_id(@RequestParam("email") String memberEmail,
    Model model) {

    String result = memberService.findById(memberEmail);

    return result;
}
```

ajax 'Post' 방식으로
Controller에
data 넘긴 후 코드
실행 가능 구현함
서버에서 data
(memberEmail)을
@RequestParam으로 받음

XML

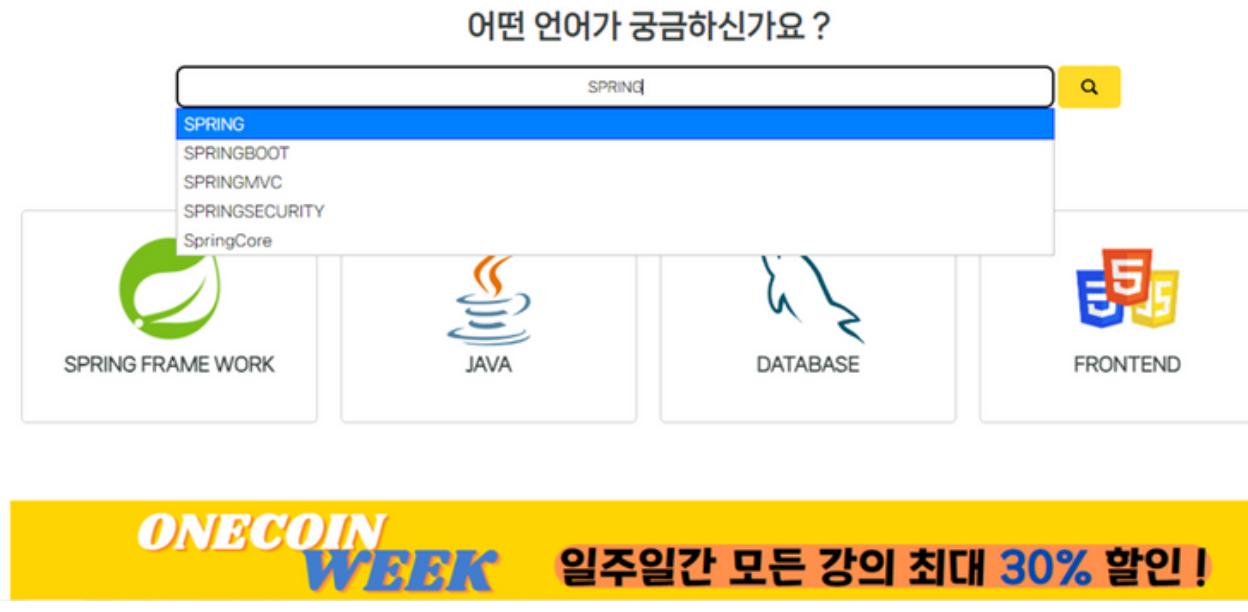
```
<select id="findId" parameterType="String" resultType="String">
    select NVL(memberId, 0) from
    member_table where memberEmail = #{memberEmail}
</select>
```

03

프로젝트 상세

3-3. 기술 설명

강의 검색 기능



Controller

```
@PostMapping("/search/autocomplete.do")
public @ResponseBody Map<String, Object> autocomplete(
    @RequestParam Map<String, Object> paramMap
){
    List<Map<String, Object>> resultList = searchService.autocomplete(paramMap);
    paramMap.put("resultList", resultList);

    return paramMap;
}
```

XML

```
<select id="autocomplete" parameterType="search" resultType="search">
    SELECT * FROM search_keyword WHERE LOWER(keyword)
        LIKE lower('%'|| #{value} || '%')
</select>
```

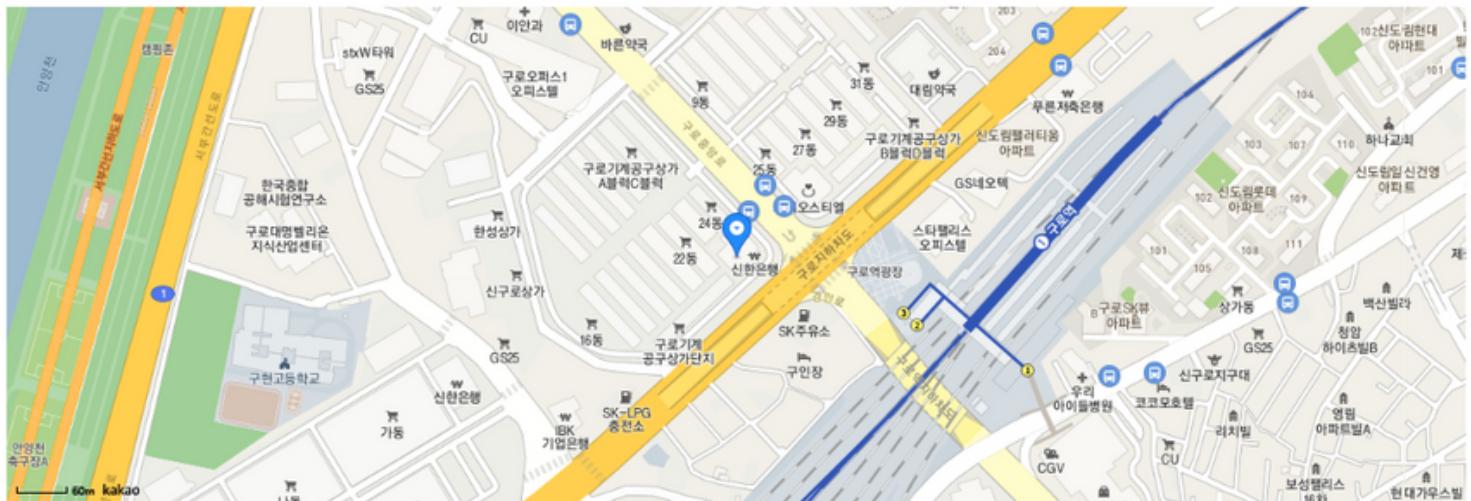
Jquery UI – autocomplete
이용하고 ajax에서 'POST'
방식으로 data 넘기고
map함수 이용하여 출력

03 프로젝트 상세

3-3. 기술 설명

찾아오는 길

주차 구역 및 안내 사항



✓ 주소 | 서울특별시 구로구 경인로 557 삼영빌딩 4층

✓ 연락처 | 02-3667-0008

지하철
구로역 1호선 하차 후 도보 5분

버스
571번, 654번, 5630번, 6613번
구로역, 구로기계공구상가 하차

주차
상가 건물 지하주차장 이용
주차장 이용 : 유료 (학원 방문 시 2시간 무료)

KaKao API로
지도, 마커를 구현하여
찾아오는길 구현

<script>

```
var mapContainer = document.getElementById('map'),
    mapOption = {
        center: new kakao.maps.LatLng(37.5030698, 126.8789856),
        level: 3
    };

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

var markerPosition = new kakao.maps.LatLng(37.5030698, 126.8789856);

window.addEventListener('resize', function() {
    map.setCenter(new kakao.maps.LatLng(37.5030698, 126.8789856));
});

var marker = new kakao.maps.Marker({
    position: markerPosition
});

marker.setMap(map);
</script>
```

Web

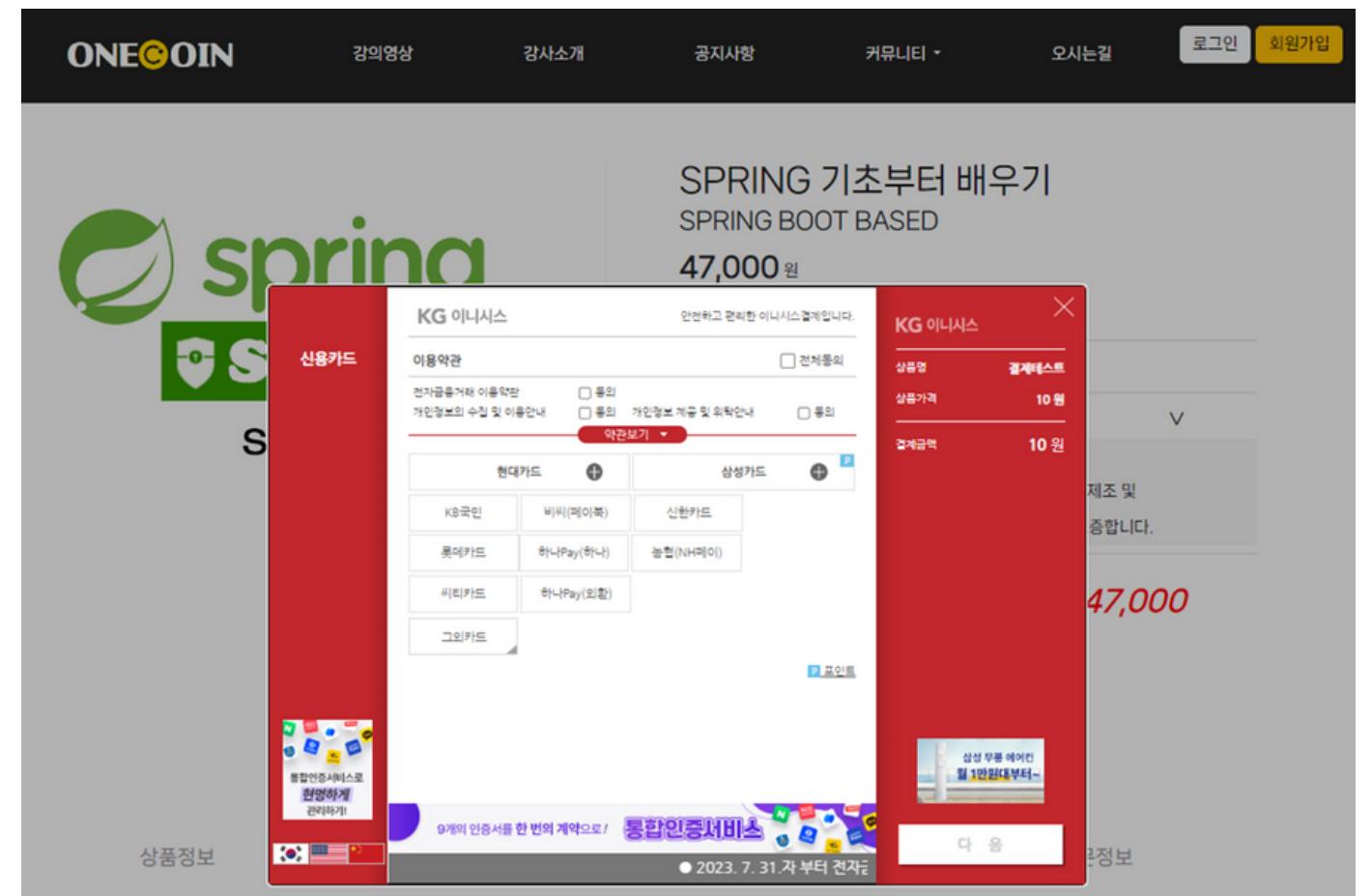
사이트 도메인

<http://192.168.0.48>

프로젝트 상세

3-3. 기술 설명

강의구매 (카드결제)



| □ | imp_uid(클릭하면 영수증보기)
merchant_uid | 결제금액
(최종환불금액) | 구분
(클릭) | PG사
(상점아이디) | PG승인번호
(카드사승인번호) | 결제상세 | 주문명
구매자(이름/이메일/휴대폰/주소) | ○ 결제시각 | 상태 |
|---|---|----------------------------------|--------------------------------------|--|--|---------------|--|---------------------|---|
| 1 | imp_614337218221
merchant_1689936337443

웹훅로그 부가 데이터 | 결제완료
₩10
(₩0) | pc
신용카드 | 이니시스(웹표준)
(INIBILLTst) | StdpayCARDINIBILLTst20230721194609028394
(36089569)
카드번호 : 621003***** | 비씨카드
(월시플) | 결제테스트
장나라
woehd1019@naver.com
010-1234-5678
[123-456]
서울특별시 강남구 삼성동 | 2023-07-21 19:46:10 | 결제완료
취소하기 |

```
// XML
<insert id="payinsert">
    insert into payMent values(#{imp_uid}, #{merchant_uid},
    #{paid_amount}, #{apply_num}, #{paid_at})
</insert>
// SCRIPT
<script>
function fn_buy() {
    var IMP = window.IMP;
    <!-- https://github.com/iamport/iamport-manual/blob/master/인증결제/sample/inicis.md -->
    IMP.init('imp68800872'); // portone 대신 자신의 "가맹점 식별코드"를 사용하시면 됩니다
    // IMP.request_pay(param, callback) 결제창 호출
    IMP.request_pay({
        pg : "html5_inicis",
        pay_method : "card",
        merchant_uid : 'merchant_' + new Date().getTime(),
        name : '결제테스트', // 상품명
        amount : 10, // 금액
        buyer_email : 'woehd1019@naver.com', // 자기이메일
        buyer_name : '장나라',
        buyer_tel : '010-1234-5678',
        buyer_addr : '서울특별시 강남구 삼성동',
        buyer_postcode : '123-456',
    }, function(rsp) { // callback
        if ( rsp.success ) {
            var paymentInfo = {
                imp_uid : rsp.imp_uid,
                merchant_uid : rsp.merchant_uid,
                paid_amount : rsp.paid_amount,
                apply_num : rsp.apply_num,
                paid_at : new Date()
            };
            $.ajax({
                url : "/productdetails/paymentProcess.do",
                method: "post",
                contentType : "application/json",
                data : JSON.stringify(paymentInfo),
                success : function(data, textStatus){
                    console.log(paymentInfo);
                    alert("결제가 완료되었습니다.");
                },
                error : function(e){
                    console.log(e);
                }
            });
        } else {
            alert("결제실패 : " + rsp.error_msg )
        }
    });
}
</script>
```

IIMPORT를
이용하여 script 구현
ajax 'POST'
방식으로 Controller에
데이터 전송 하여
결제 구현

03 프로젝트 상세

3-3. 기술 설명

글 등록 기능

세상의 모든 오류 여기있습니다

게시판 등록

여기에 제목을 적으세요

작성자를 기입하세요

내용을 입력하세요

등록하기!!!

Controller

```
@PostMapping("/yea_enroll")
public String boardEnrollPOST(Yea_BoardVO board, RedirectAttributes rttr) {
    log.info("Yea_BoardVO : " + board);
    bservice.enroll(board);
    rttr.addFlashAttribute("result", "enroll success");
    return "redirect:/board/yea_list";
}
```

XML

```
<insert id="enroll">
    insert into yea_board(bno, title, content, writer)
    values (yea_seq_board.nextval, #{title}, #{content}, #{writer})
</insert>
```

mapper.xml에 <insert>태그
id 속성 값을 메서드 이름과
동일하게 작성하여 글 등록 처리가
완료되면 addFlashAttribute에
저장 후 yea_list로 돌아가게
redirect 처리

글 조회 기능

게시판 번호	40
게시판 제목	첫번째 글등록 제목입니다.
게시판 내용	첫번째 글등록 내용입니다.
게시판 작성자	ONECOIN
게시판 등록일	2023/07/21
게시판 수정일	2023/07/21
목록 페이지 수정하기	

Controller

```
@GetMapping("/yea_get")
public void boardGetPageGET(int bno, Model model, Yea_Criteria cri) {
    model.addAttribute("pageInfo", bservice.getPage(bno));
    model.addAttribute("cri", cri);
}
```

bno로 조회 할 수 있는 쿼리문을
작성하고 해당의 게시글을 불러옴

XML

```
<select id="getPage" resultType="org.zerock.model.Yea_BoardVO">
    select * from yea_board where bno = #{bno}
</select>
```

03 프로젝트 상세

3-3. 기술 설명

글 수정 가능

게시판 번호
40

게시판 제목
첫번째 글등록 수정 제목입니다.

게시판 내용
첫번째 글등록 수정 내용입니다.

게시판 작성자
ONECOIN

게시판 등록일 | 2023/07/21 | 게시판 수정일 | 2023/07/21

[목록 페이지](#) [수정 완료](#) [수정 취소](#) [삭제](#)

게시판 번호 | 40
게시판 제목 | 첫번째 글등록 수정 제목입니다.
게시판 내용 | 첫번째 글등록 수정 내용입니다.
게시판 작성자 | ONECOIN
게시판 등록일 | 2023/07/21 | 게시판 수정일 | 2023/07/21
[목록 페이지](#) [수정 하기](#)

Controller

```
@GetMapping("/yea_modify")
public void boardModifyGET(int bno, Model model, Yea_Criteria cri) {
    model.addAttribute("pageInfo", bservice.getPage(bno));
    model.addAttribute("cri", cri);
}
```

XML

```
<update id="modify">
    update yea_board set title = #{title},
    content =#{content}, updatedate = sysdate where bno = #{bno}
</update>
```

수정하고자 하는 게시글의 내용을
출력해야 하기 때문에 int형
파라미터와 해당 게시글의 내용
을 호출하는 Service 메소드
(getPage())를 호출

03

프로젝트 상세

3-3. 기술 설명

글 삭제 기능

Oyeo 게시판

오류를 알려주세요!

게시글 목록				
번호	제목	작성자	작성일	수정일
41	두번째제목	ONECOIN	2023/07/21	2023/07/21
40	첫번째 글등록 수정 제목입니다.	ONECOIN	2023/07/21	2023/07/21

주제의 경색어를 입력하세요.

ONECOIN

localhost 내용:
삭제가 완료되었습니다.

오류를 알려주세요!

게시글 목록				
번호	제목	작성자	작성일	수정일
41	두번째제목	ONECOIN	2023/07/21	2023/07/21

모시는길 로그인 회원가입 확인

Controller

```
@PostMapping("/yea_delete")
public String boardDeletePOST(int bno, RedirectAttributes rttr) {
    bservice.delete(bno);
    rttr.addFlashAttribute("result", "delete success");
    return "redirect:/board/yea_list";
}
```

선택한 bno의 게시글을
삭제하면 알림창을 띠움

XML

```
<delete id="delete">
    delete from yea_board where bno = #{bno}
</delete>
```

댓글 등록 가능

• 댓글목록

댓글달기

첫번째댓글

댓글을 달아봅니다

14:47:44

```
@PostMapping(value = "/new",
    consumes = "application/json",
    produces = {MediaType.TEXT_PLAIN_VALUE})
public ResponseEntity<String> create(@RequestBody YooReplyVO vo){
    log.info("YooReplyVO 댓글 객체 : " + vo);

    int insertCount = service.register(vo);
    log.info("등록된 댓글 개수는 = " + insertCount);

    return insertCount == 1
        ? new ResponseEntity<String>("success", HttpStatus.OK)
        : new ResponseEntity<String>(HttpStatus.INTERNAL_SERVER_ERROR);
}
```

Controller

```
<insert id="insert">
    INSERT into tbl_reply (rno, bno, reply, replyer)
    VALUES (seq_reply.nextval, #{bno}, #{reply}, #{replyer})
</insert>
```

XML

consumes와 produces를 이용해서 JSON 방식의 데이터만 처리하고,
@RequestBody를 적용해서 JSON 데이터를 YooReplyVO로 변환하도록 지정

댓글 목록 기능

댓글목록

댓글달기

첫번째댓글

댓글을 달아봅니다

14:47:44

```
@GetMapping(value = "/pages/{bno}/{page}",
    produces = {
        MediaType.APPLICATION_XML_VALUE,
        MediaType.APPLICATION_JSON_UTF8_VALUE})
public ResponseEntity<List<YooReplyVO>> getList(
    @PathVariable("page") int page,
    @PathVariable("bno") Long bno){
    Log.info("특정 게시글에 등록된 댓글의 리스트 목록 내용 확인!");
    YooCriteria cri = new YooCriteria(page, 10);
    Log.info(cri);
    return new ResponseEntity<List<YooReplyVO>>(service.getList(cri, bno), HttpStatus.OK);
}
```

Controller

댓글 목록을 확인하는 getList()
메서드를 선언하고 게시물의
번호는 @PathVariable을
이용해서 파라미터로 처리

XML

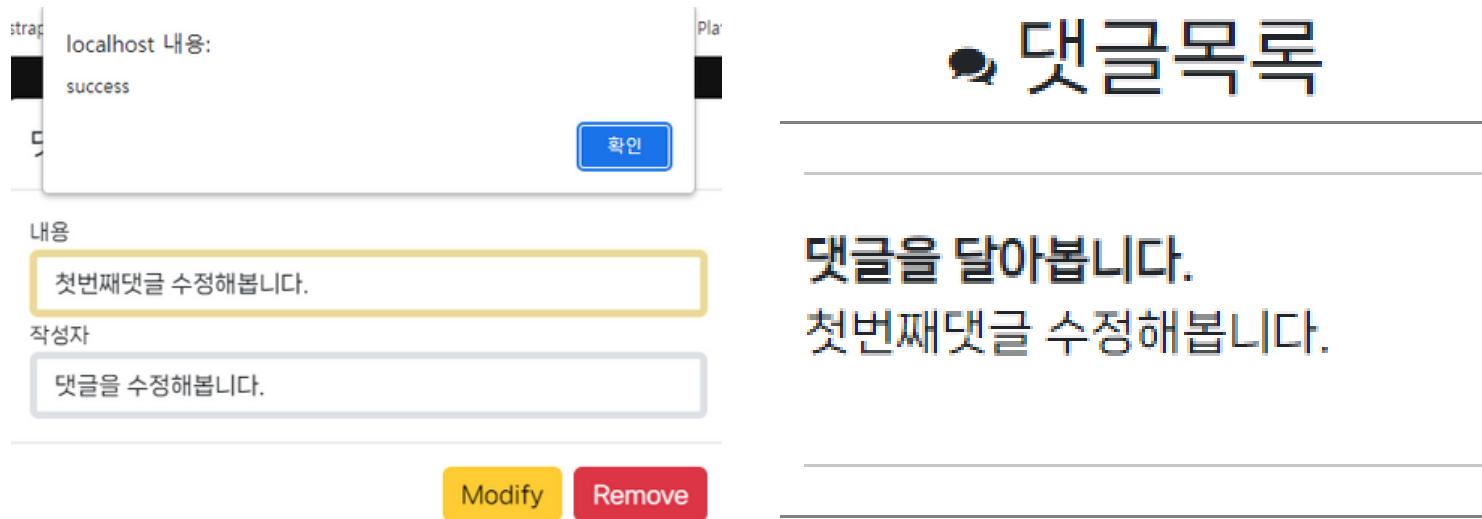
```
<select id="read" resultType="org.zerock.domain.YooReplyVO">
    select * from tbl_reply where rno = #{rno}
</select>
```

03

프로젝트 상세

3-3. 기술 설명

댓글 수정 가능



댓글목록

댓글을 달아봅니다.

첫번째댓글 수정해봅니다.

```
@RequestMapping(method = {RequestMethod.PUT, RequestMethod.PATCH},
    value = "/modify/{rno}",
    consumes = "application/json",
    produces = {MediaType.TEXT_PLAIN_VALUE})
public ResponseEntity<String> modify(
    @RequestBody YooReplyVO vo,
    @PathVariable("rno") Long rno){
    vo.setRno(rno);
    log.info("rno 댓글 번호 = " + rno);
    log.info("modify 수정처리 = " + vo);

    return service.modify(vo) == 1
        ? new ResponseEntity<String>("success", HttpStatus.OK)
        : new ResponseEntity<String>(HttpStatus.INTERNAL_SERVER_ERROR);
}
```

Controller

```
<update id="update">
UPDATE tbl_reply set reply = #{reply}, updatedate = sysdate
WHERE rno = #{rno}
</update>
```

XML

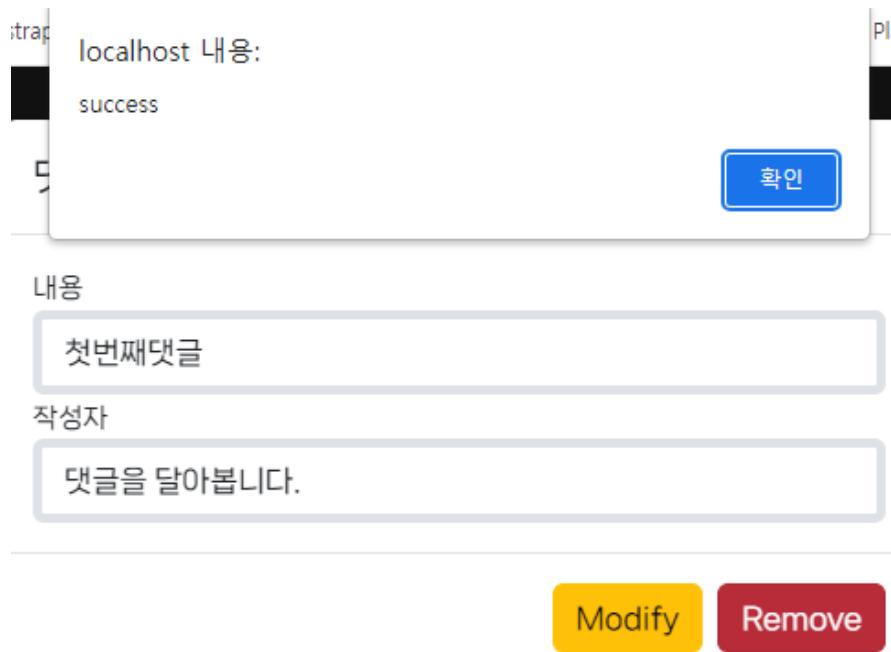
JSON 데이터 형식이므로
@RequestBody를 이용해서
처리하고,
리소스를 업데이트 하는 PUT과
PATCH를 사용

03

프로젝트 상세

3-3. 기술 설명

댓글 삭제 기능



```

@Controller
@RequestMapping(value = "/delete/{rno}",
    consumes = "application/json",
    produces = {MediaType.TEXT_PLAIN_VALUE})
public ResponseEntity<String> remove(@PathVariable("rno") Long rno){
    log.info("댓글 삭제 처리 : " + rno);

    return service.remove(rno) == 1
        ? new ResponseEntity<String>("success", HttpStatus.OK)
        : new ResponseEntity<String>(HttpStatus.INTERNAL_SERVER_ERROR);
}

```

Controller

rno와 맞는 댓글 삭제
삼항 연산자를 이용해
true => success
false -> error
출력

```

<delete id="delete">
    DELETE from tbl_reply WHERE rno = #{rno}
</delete>

```

XML

03 프로젝트 상세

3-3. 기술 설명

SPRINGBOOT / REACT 게시판

자유게시판

×

게시글 등록

최근 게시글 보기

최근 게시글 수정

최근 게시글 삭제

게시글 목록 보기

```
crud(){
  const {articleId, articleTitle, articleContent, crud} = this.state;

  let crudType = "";
  if(crud === "Update"){
    crudType = "/updateProcess.do";
  } else if(crud === "Delete"){
    crudType = "/deleteProcess.do";
  } else if(crud === "Insert"){
    crudType = "/insertProcess.do";
  } else if(crud === "View"){
    return null;
  } else if(crud === "boardList"){
    return BoardList;
  }

  let form = new FormData();
  form.append("articleContent", articleContent);
  form.append("articleTitle", articleTitle);
  if(crud !== "Insert"){
    form.append("articleId", articleId);
  }
}
```

Controller

```
@RestController
@CrossOrigin(origins = "http://192.168.0.48:3000/")
public class ArticleController {

  @Autowired
  private ArticleService articleService;

  @RequestMapping(value = "insertProcess.do")
  public void insertProcess(Article article) {
    articleService.insert(article);
  }

  @RequestMapping(value = "updateProcess.do")
  public void updateProcess(Article article) {
    articleService.update(article);
  }

  @RequestMapping(value = "deleteProcess.do")
  public void deleteProcess(Article article) {
    articleService.delete(article);
  }

  @RequestMapping(value = "view.do")
  public Article viewArticle() {
    Article article = articleService.select();
    return article;
  }

  @RequestMapping(value = "list.do")
  public List<Article> findAll() {
    return articleService.findAll();
  }
}
```

REACT의 Axios, Proxy와
SPRINGBOOT, OracleDB 연결 후
Controller에 Mapping된 url로 이동

03

프로젝트 상세

3-3. 기술 설명

게시글 등록

안녕하세요!

내용
안녕하세요!!

확인

게시글 등록

취소

게시글 조회

안녕하세요!

내용
안녕하세요!!

확인

게시글 수정

안녕하세요!

내용
안녕하세요!! 흐흐

확인

게시글 수정

취소

게시글 삭제

hi

hi

게시글 삭제

취소

SPRINGBOOT / REACT 게시판 CRUD

```

@Service("articleService")
public class ArticleServiceImpl implements ArticleService {

    @Autowired
    private ArticleRepository dao;

    @Override
    public void insert(Article article) {
        dao.save(article);
    }

    @Override
    public void update(Article article) {
        Article currArticle = dao.findTopByOrderByIdDesc();
        currArticle.setArticleTitle(article.getArticleTitle());
        currArticle.setArticleContent(article.getArticleContent());
        dao.save(currArticle);
    }

    @Override
    public void delete(Article article) {
        dao.delete(article);
    }

    @Override
    public Article select() {
        return dao.findTopByOrderByIdDesc();
    }
}

```

ServiceImpl

REACT의 Axios, Proxy와
SPRINGBOOT, OracleDB 연결 후
if문을 사용하여
해당하는 crud를 실행

```

createHeaderName(){
    const crud = this.state.crud;
    if(crud === "View"){
        return "조회";
    } else if(crud === "Update"){
        return "수정";
    } else if(crud === "Delete"){
        return "삭제";
    } else if(crud === "Insert"){
        return "등록";
    }
}

createCrudBtn(){
    const crud = this.state.crud;
    if(crud === "View"){
        return null;
    } else {
        const crudName =
            crud === "Update" ? "수정" : crud === "Insert" ? "등록" : "삭제";
        return(
            <Button variant="outline-dark" size="lg" className="main_btn" onClick={() => this.crud()}>게시글 {crudName}</Button>
        );
    }
}

```

03 프로젝트 상세

3-3. 기술 설명

SPRINGBOOT / REACT 게시판 목록

```

@Data
@Entity
@Table(name = "springboot_crud")
public class Article {

    @Id
    @GeneratedValue
    @Column(name = "article_id")
    private int articleId;

    @Column(name = "article_title")
    private String articleTitle;

    @Column(name = "article_content")
    private String articleContent;
}
  
```

```

<h2 className='text-center page_top'>게시글 목록 조회</h2>
<div className='row'>
  <Table striped bordered hover className='table_list'>
    <thead>
      <tr>
        <th>글번호</th>
        <th>글제목</th>
        <th>글내용</th>
      </tr>
    </thead>
    <tbody>
      {
        this.state.boards.map(
          board =>
          <tr key = {board.articleId}>
            <td>{board.articleId}</td>
            <td>{board.articleTitle}</td>
            <td>{board.articleContent}</td>
          </tr>
        )
      }
    </tbody>
  </Table>
</div>
  
```

REACT의 Axios, Proxy와
SPRINGBOOT, OracleDB
사용해 연결 후
해당 컬럼을 받아 목록으로 출력

글번호	글제목	글내용
101	안녕하세요	안녕하세요

03

프로젝트 상세

3-3. 기술 설명

```

@CrossOrigin
@Controller
public class MemberController {

    @Resource(name = "memberService")
    private MemberService memberService;

    @RequestMapping(value = "/loginAction.do")
    public @ResponseBody String loginAction
        (@ModelAttribute("memberVO")MemberVO memberVO,
         HttpServletRequest request, HttpServletResponse response,
         HttpSession session, Model model)
        throws Exception{
        String result = "N";

    try {
        MemberVO resultVO = memberService.selectMemberByIdByPw(memberVO);
        if(resultVO != null) {
            result = resultVO.getMemberName();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return result;
}

```

localhost:3000 내용:
장나라 회원님, 환영합니다!

Controller

SPRINGBOOT / REACT 로그인

```

if(!submitYN){
    var submitYN = true;

    const form = new FormData()
    form.append('memberId', memberId)
    form.append('memberPw', memberPw)

    console.log("click login");

    axios.post("http://192.168.0.48:9008/loginAction.do",
    form,{
        Headers:{
            'Content-type' : 'application/json',
        }
    })
    .then(res) => {
        console.log(res.data);
        if(res.data != "N"){
            alert(res.data + " 회원님, 환영합니다!");
            sessionStorage.setItem("MEMBERID", memberId);
            sessionStorage.setItem("name", res.data);
            document.location.href="/";
        }else{
            alert("회원 정보가 없습니다!");
        }
    })
    .catch()
}

```

Springboot, Maven,
OracleDB,Axios 이용해
변수 값을 비교하여
true => 페이지 이동
false => 회원 정보 X

로그인

jang

....

로그인

03

프로젝트 상세

3-3. 기술 설명

SPRINGBOOT / REACT 로그아웃

```

const [isLogin, setIsLogin] = useState(false);
const name = sessionStorage.name
// 로그인 상태 관리 처리
useEffect(() =>
  if(sessionStorage.getItem('MEMBERID') === null){
    console.log('isLogin 상태 = ', isLogin);
  } else{
    // 로그인 상태 변경
    setIsLogin(true)
    console.log("isLogin 상태 = ", isLogin);
  }
);

const onLogout = () => {
  alert("로그아웃 됨");
  // sessionStorage에 저장된 Item 삭제처리
  sessionStorage.removeItem('MEMBERID')
  sessionStorage.removeItem('name')

  // '/' url로 이동
  document.location.href = '/'
}
  
```

The screenshots show the ONECOIN login page. The top one shows the '로그인 하러 가기' button, and the bottom one shows the '로그아웃 하러 가기' button. This visualizes the state change from logged in to logged out.

sessionStorage 이용
DB에 있는 MEMBERID가
'null'이 아닐 때

로그인 상태를 true

로그아웃 시
sessionStorage에 있는
데이터를 삭제
isLogin이
"true" => 로그아웃
"false" => 로그인
출력

```

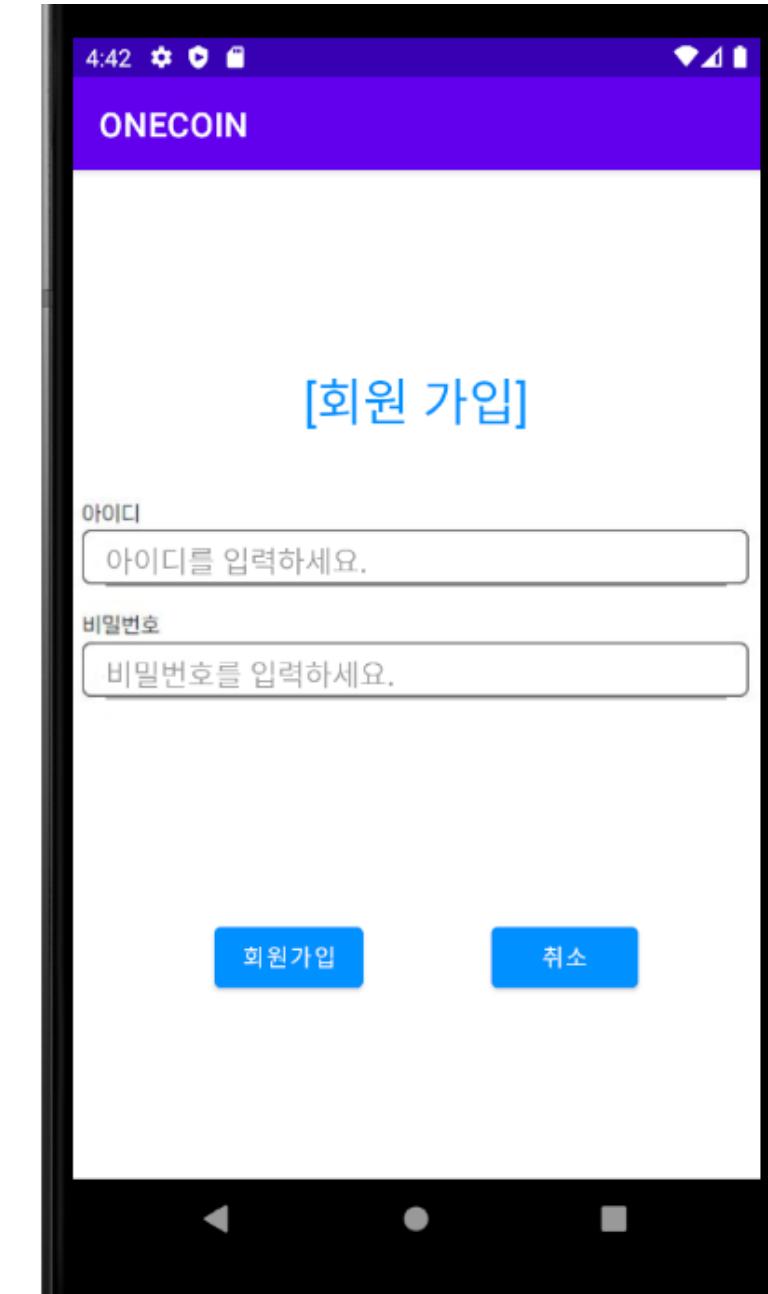
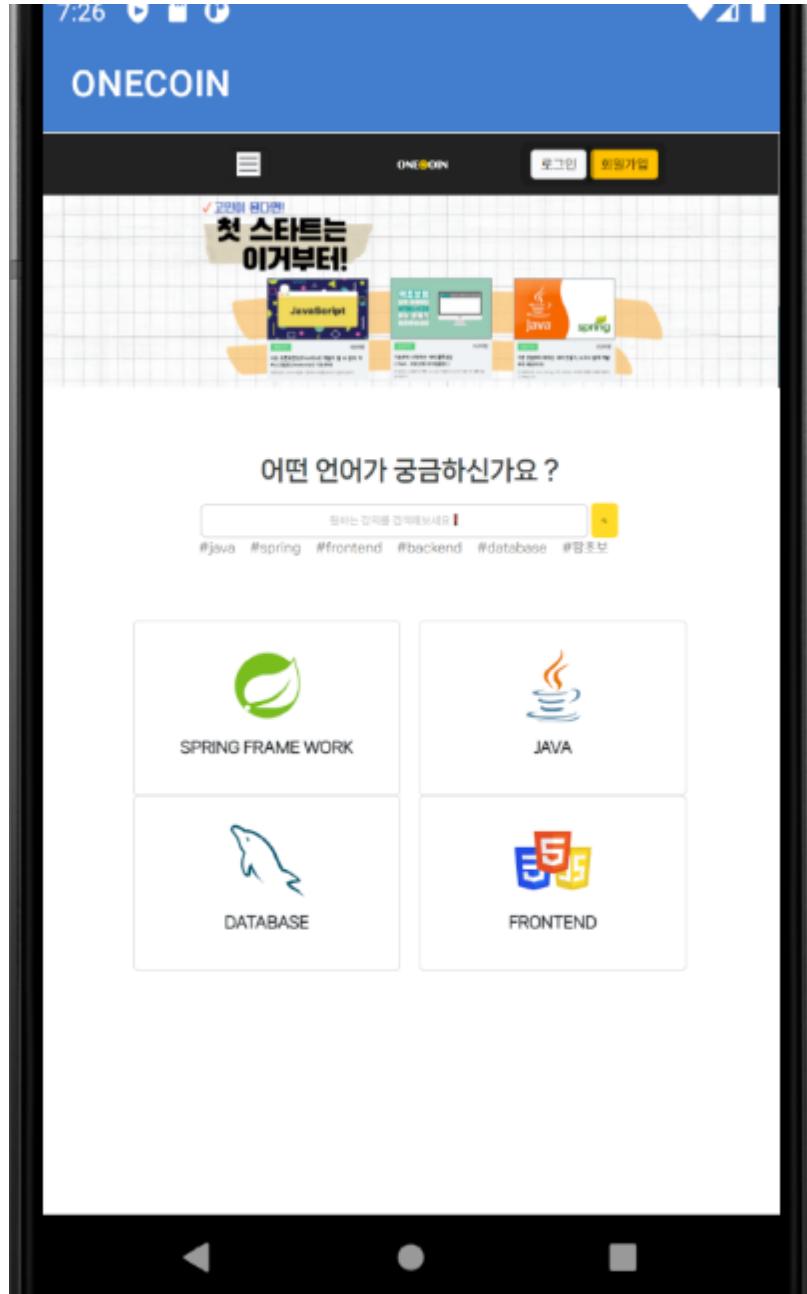
<Link to = "/login">
{isLogin
 ?
<Button variant="outline-warning" className="login_btn" onClick={onLogout}>로그아웃 하러 가기</Button>
:
<Button variant="outline-warning" className="login_btn">로그인 하러 가기</Button>
}
</Link>
  
```

03

프로젝트 상세

3-3. 기술 설명

ADROID / SPRINGBOOT 메인 페이지



ADROID / SPRINGBOOT 메인 페이지

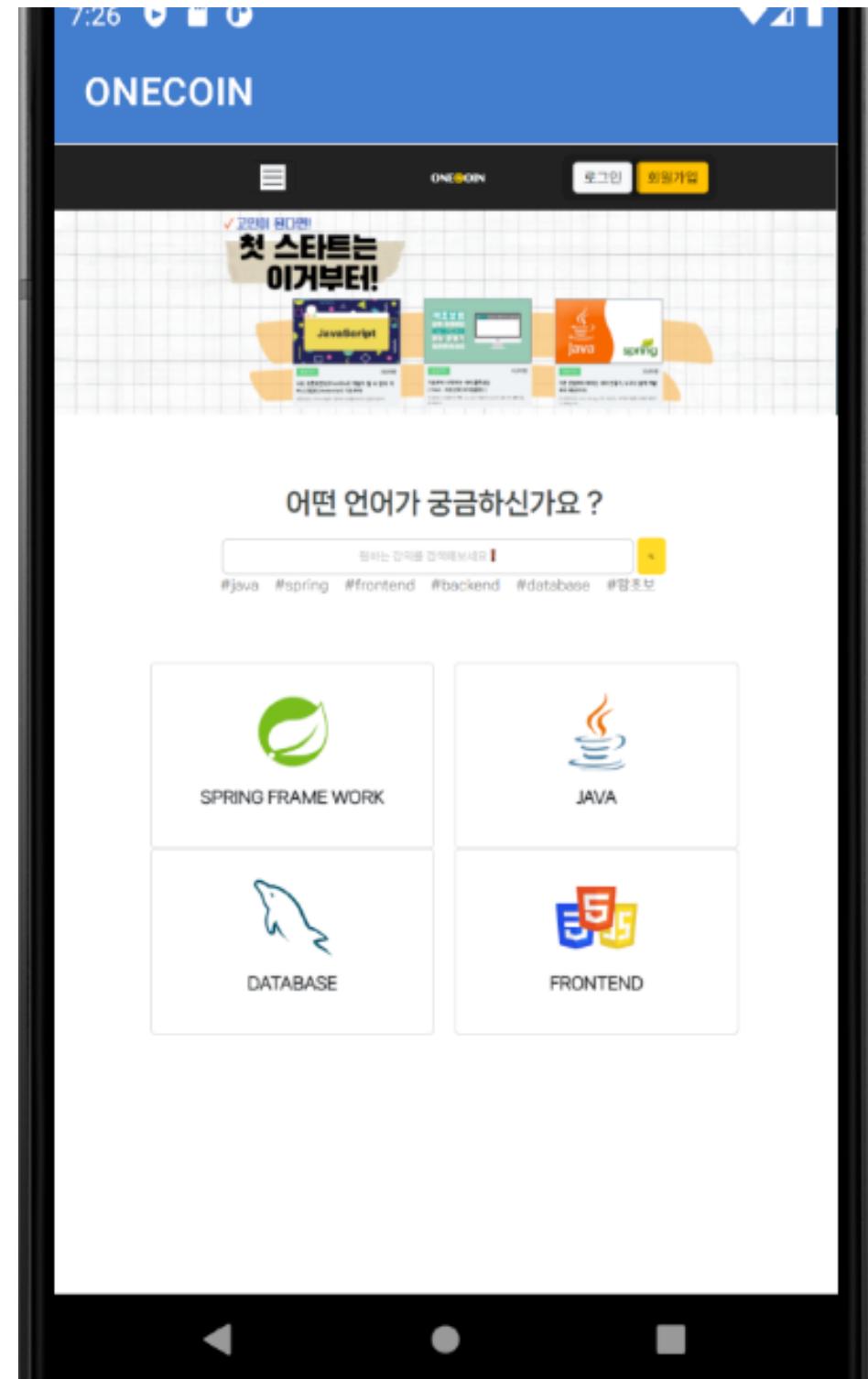
```
public class HomeActivity extends AppCompatActivity {

    Button button, button2;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);

        button = (Button)findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(),MainActivity.class);
                startActivity(intent);
            }
        });

        button2 = (Button)findViewById(R.id.button2);
        button2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(),SignUpActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

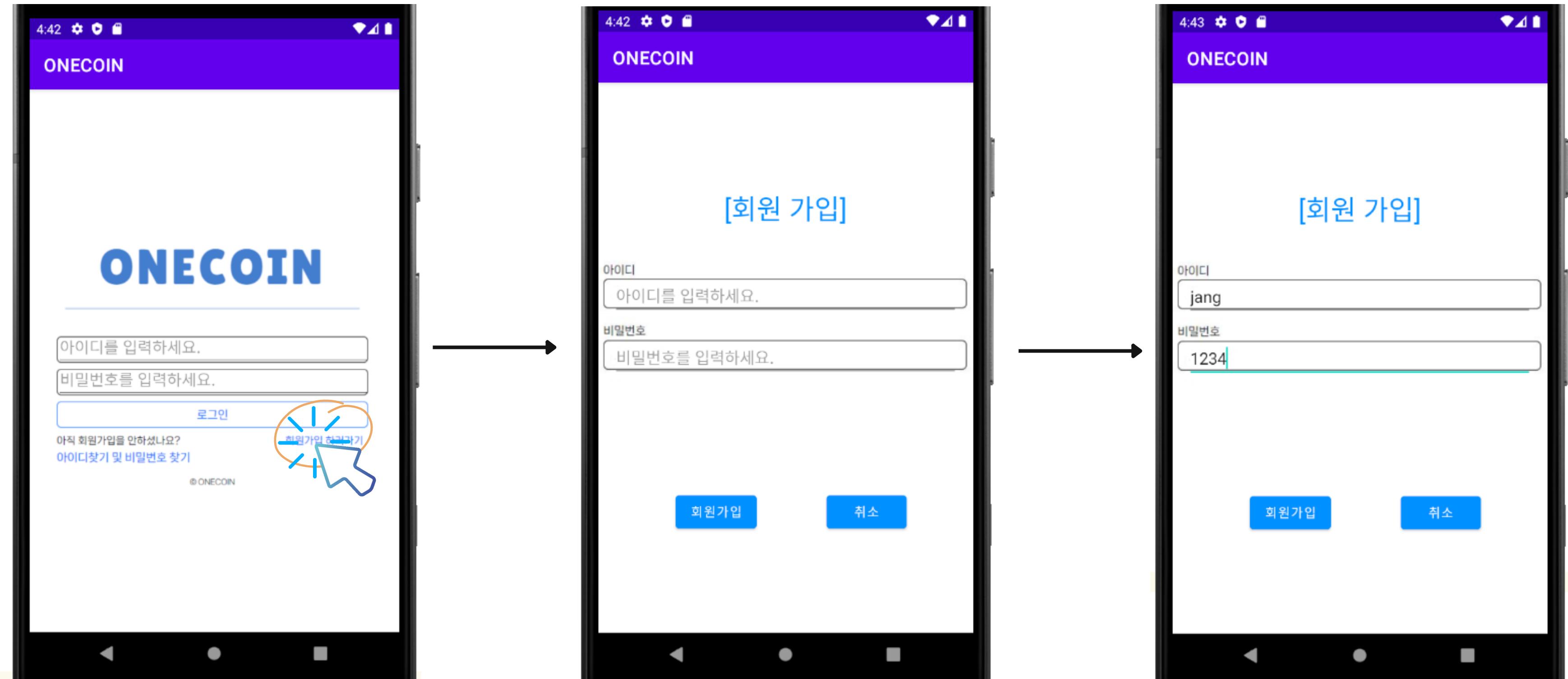


안드로이드 앱 실행 시
HomeActivity 열림
로그인 버튼 클릭 시 로그인 창
회원가입 버튼 클릭 시 회원가입
창으로 이동

03

프로젝트 상세 3-3. 기술 설명

ADROID / SPRINGBOOT 회원가입



03

프로젝트 상세

3-3. 기술 설명

```

@Entity
@Table(name = "user_app")
@Getter
@Setter
@ToString
public class User {

    @Id
    @Column(name = "user_id")
    private String userId;

    @Column(name = "user_password")
    private String userPassword;
}

public class User {
    String userId;
    String userPassword;

    public String getUserId() {
        return userId;
    }

    public void setUserId(String userId) { this.userId = userId; }

    public String getUserPassword() { return userPassword; }

    public void setUserPassword(String userPassword) { this.userPassword = userPassword; }
}

```

Retrofit

```

public class SignUpActivity extends AppCompatActivity {

    private final String TAG = getClass().getSimpleName();

    Button btnCancel, btnSignUp;
    EditText idInput, pwInput;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_up);

        idInput = (EditText)findViewById(R.id.idInput);
        pwInput = (EditText)findViewById(R.id.pwInput);

        btnCancel = (Button)findViewById(R.id.btnCancel);
        btnCancel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), HomeActivity.class);
                startActivity(intent);
            }
        });
    }
}

```

```

@RequestMapping(value = "/signup")
public boolean signUp(@RequestBody User vo) {
    System.out.println(vo);

    try {
        service.insert(vo);
        System.out.println(true);
        return true;
    } catch (Exception e) {
        System.out.println(false);
        return false;
    }
}

```

```

btnSignUp = (Button)findViewById(R.id.btnSignUp);
btnSignUp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String id = idInput.getText().toString();
        String pw = pwInput.getText().toString();

        User signUpInfo = new User();
        signUpInfo.setUserId(id);
        signUpInfo.setUserPassword(pw);

        Moshi moshi = new Moshi.Builder().build();
        Retrofit retrofit = new Retrofit.Builder()
            .baseUrl("http://192.168.0.44:9003/")
            .addConverterFactory(MoshiConverterFactory.create(moshi))
            .build();

        UserApi api = retrofit.create(UserApi.class);

        Call<Boolean> call = api.createUser(signUpInfo);
        call.enqueue(new Callback<Boolean>() {
            @Override
            public void onResponse(Call<Boolean> call, Response<Boolean> response) {
                boolean signUpSuccess = response.body();

                if (signUpSuccess){
                    Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                    startActivity(intent);
                } else {
                    Toast.makeText(getApplicationContext(), text: "회원가입에 실패했습니다!", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<Boolean> call, Throwable t) {
                Log.d(TAG, msg: "Fail msg : " + t.getMessage());
                Toast.makeText(getApplicationContext(), text: "회원가입에 실패했습니다!", Toast.LENGTH_SHORT).show();
            }
        });
    }
});

```

Retrofit을 이용하여 변환된 인터페이스를
변수에 담은 후
Call 메서드를 이용하여
받은 데이터를 if 문에 담아 회원가입 처리

ADROID / SPRINGBOOT 로그인



03 프로젝트 상세

3-3. 기술 설명

ADROID / SPRINGBOOT 로그인

```

@Controller
@RestController
public class UserController {

    @Autowired
    private UserService service;

    @RequestMapping(value = "/signin")
    public boolean signIn(@RequestBody User vo, HttpServletRequest request) {
        User user;
        try {
            user = service.select(vo);
        } catch (NullPointerException e) {
            user = null;
        }
        if (user != null) {
            if(user.getUserPassword().equals(vo.getUserPassword())){
                return true;
            }
        }
        return false;
    }
}

```

Controller

```

public class User {
    String userId;
    String userPassword;

    public String getUserId() {
        return userId;
    }

    public void setUserId(String userId) { this.userId = userId; }

    public String getUserPassword() { return userPassword; }

    public void setPassword(String userPassword) { this.userPassword = userPassword; }
}

```

Retrofit

```

Retrofit retrofit = new Retrofit.Builder()
    .baseUrl("http://192.168.0.44:9003/")
    .addConverterFactory(MoshiConverterFactory.create(moshi))
    .build();

UserApi api = retrofit.create(UserApi.class);

Call<Boolean> call = api.postUser(loginInfo);
call.enqueue(new Callback<Boolean>() {
    @Override
    public void onResponse(Call<Boolean> call, Response<Boolean> response) {
        if(response.isSuccessful()){
            boolean userExists = response.body();

            if(userExists){
                Intent intent = new Intent(getApplicationContext(), SingUpSuccess.class);
                startActivity(intent);
            } else {
                Toast.makeText(getApplicationContext(), text: "아이디 혹은 비밀번호가 일치하지 않습니다!", Toast.LENGTH_SHORT).show();
            }
        }
    }

    @Override
    public void onFailure(Call<Boolean> call, Throwable t) {
        Log.d(TAG, msg: "Fail msg : " + t.getMessage());
    }
});

@Entity
@Table(name = "user_app")
@Getter
@Setter
@ToString
public class User {
    @Id
    @Column(name = "user_id")
    private String userId;

    @Column(name = "user_password")
    private String userPassword;
}

```

SpringBoot User 클래스의
table 값을
AndoridStudio의 User 클래스에
연동시켜 로그인

프로젝트 시연

4-1. 시연 영상

4-2. 후기

프로젝트 시연

4-1. 시연 영상

4-2. 후기

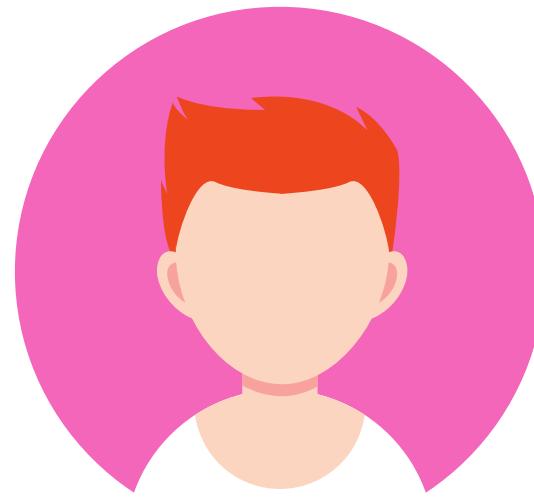
04 프로젝트 시연

4-2. 후기



유승관

실전에서의 경험이 정말 중요하다는 것을 느꼈습니다.



김재동

정말 값진 시간이었습니다



안예인

자기 능력을 향상 시킬 수 있는 시간이었습니다



차승혁

경험을 통해 많은 것을 배울 수 있는 시간이었습니다!



김정진

부족한 부분을 보완 할 수 있던 유익한 시간이었습니다

감사합니다!