



PortFolio

Marry Me?

조동현

목차

1. 프로젝트 개요
2. 개발 환경
3. WBS
4. Use Case
5. 사이트 구성
6. ERD
7. REST API
8. 핵심 기능
9. Trouble Shooting
10. 추후 작업
11. 소감

1. 프로젝트 개요

구분	내용
프로젝트명	Marry Me
개발 목적	결혼을 원하는 사람들이 간단하게 자신의 이상형을 찾을 때 이용하게 하기 위하여
대상	결혼 적령기의 남녀
주요 기능	이상형 검색, 총 회원 현황(통계) 보기, 고객 게시판 등

2. 개발 환경

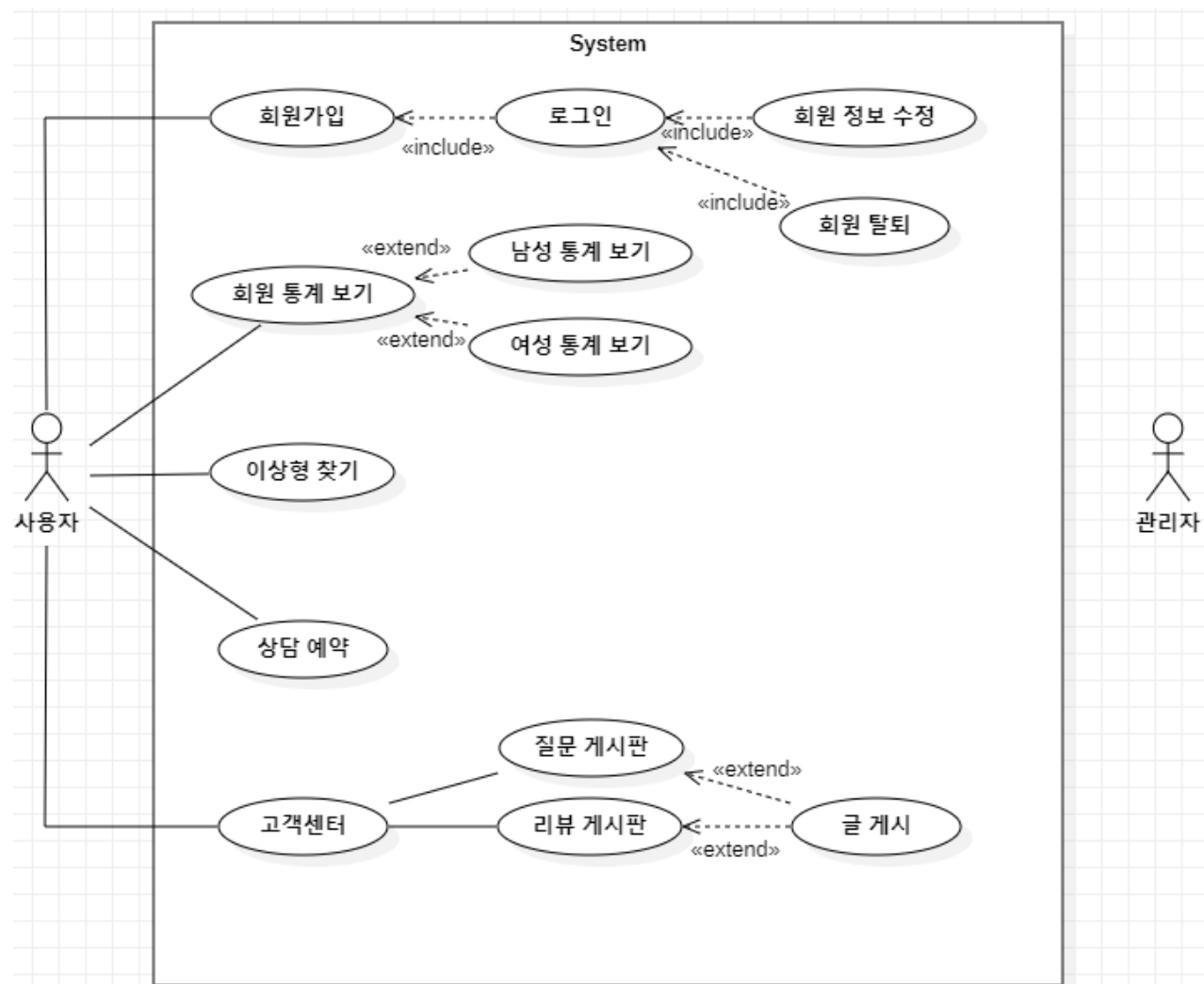
- | | |
|--------------|----------------------|
| ✓ OS(운영 체제) | Window 10 64bit |
| ✓ Language | Java 11 |
| ✓ FrameWork | Spring Boot 2.7.7 |
| ✓ Library | Lombok, JPA |
| ✓ Build Tool | Gradle |
| ✓ DBMS | MySQL 8.0 |
| ✓ View | Thymeleaf, BootStrap |



3. WBS

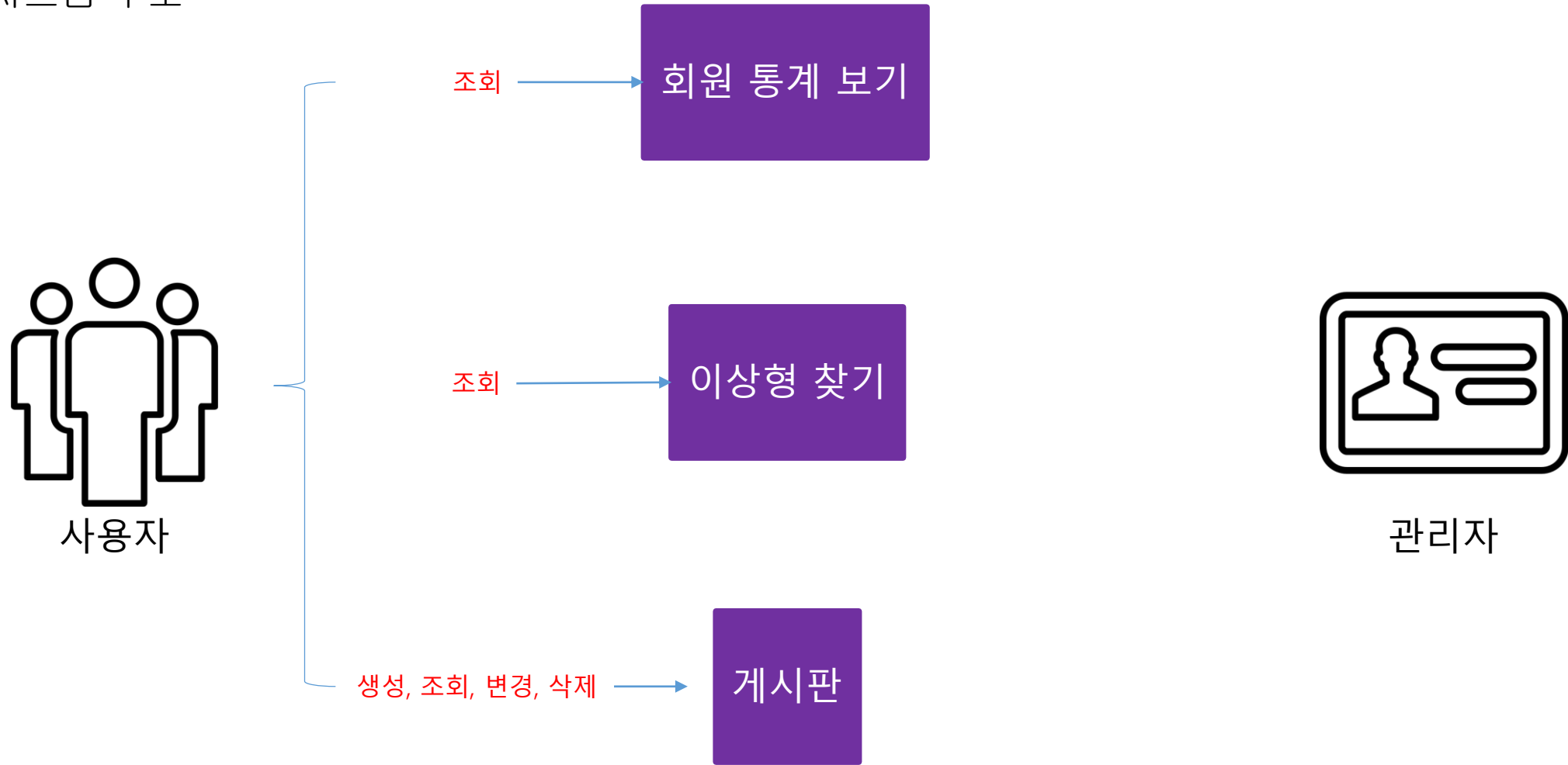
Work Break-down Structure						12월																1월																		
						3주							4주							5주							1주							2주						
Procedures	Steps	Tasks	담당자	schedule		산출물/비고	일	월	화	수	목	금	토	일	월	화	수	목	금	토	일	월	화	수	목	금	토	일	월	화	수	목	금	토	일	월	토	토	토	토
				시작일	종료일		11	12	13	14	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16				
1.0.0. 착수 및 프로젝트 관리																																								
1.1.0. 프로젝트 관리																																								
1.1.1. 프로젝트 주제 선정			조동현	12월 12일	12월 15일																																			
1.1.2. 일정 계획, 일정 정리			조동현	12월 14일	12월 15일																																			
1.2.0. 프로젝트 관리 실무																																								
1.2.1. 완료보고			조동현	12월 22일	12월 23일	완료보고																																		
1.0.0. 분석/설계																																								
1.1.0. 화면 분석/설계																																								
1.1.1. 화면 설계서 분석/제작			조동현	12월 24일	12월 25일	화면 설계서																																		
1.1.2. 화면 설계서 컨펌			조동현	12월 25일	12월 26일																																			
2.0.0. 프론트 개발																																								
2.1.0. 프론트 개발																																								
2.1.1. index 페이지			조동현	12월 26일	12월 27일																																			
2.1.2. login 페이지			조동현	12월 27일	12월 28일																																			
2.1.3. movies 페이지			조동현	12월 28일	12월 29일																																			
2.1.4. movie_detail 페이지			조동현	12월 29일	12월 30일																																			
2.1.5. Board 페이지			조동현	12월 30일	12월 31일																																			
3.0.0. 백엔드 개발																																								
3.1.0. 개발 설계부분																																								
3.1.1. ERD 제작 및 DB 생성			조동현	12월 31일	1월 1일	ERD																																		
3.1.2. Entity 생성 및 Repository class 생성			조동현	1월 1일	1월 3일																																			
3.1.3. Controller 및 Service class 생성			조동현	1월 3일	1월 5일																																			
3.1.4. view 연결			조동현	1월 5일	1월 8일																																			
4.0.0. 테스트 및 완료																																								
4.1.0. 테스트																																								
4.1.1. 발표 자료 작성			조동현	1월 8일	1월 11일																																			
4.1.2. 최종 점검			조동현	1월 11일	1월 16일																																			

4. Use Case Diagram



5. 사이트 구성

5.1 시스템 구조



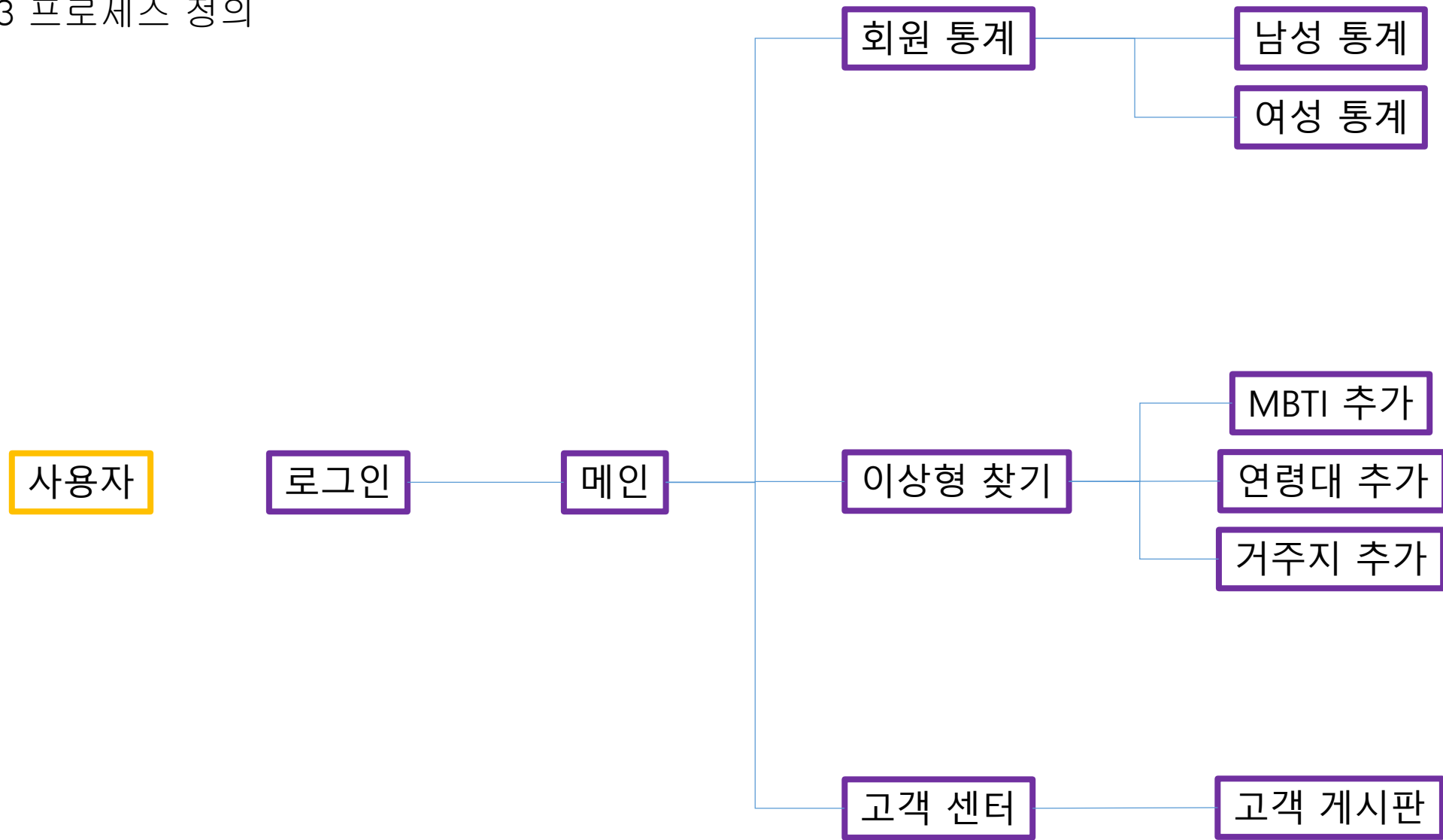
5. 사이트 구성

5.2 사이트 맵 구조

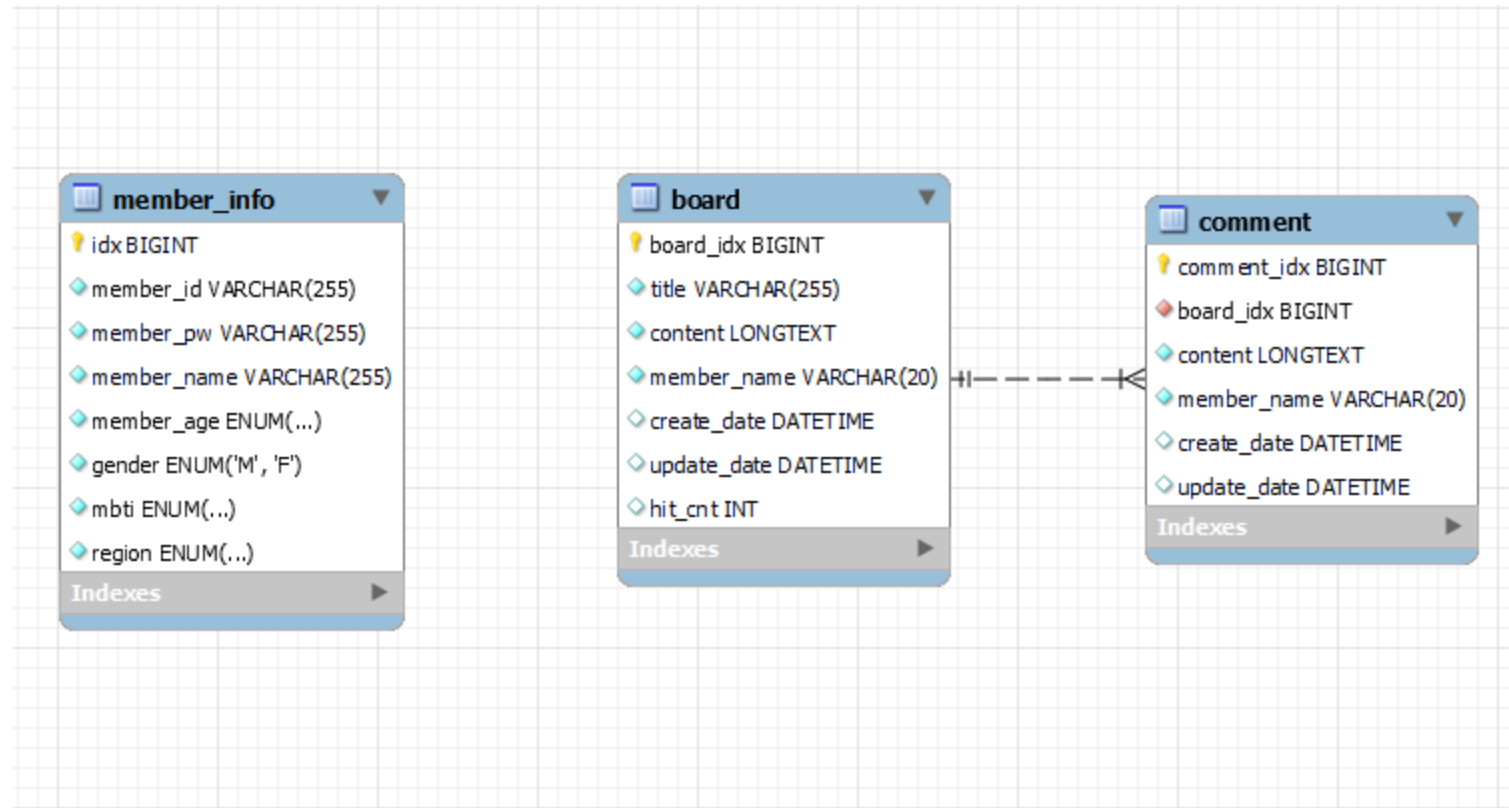
Web Site 메뉴 구조						
1depth	2depth	3depth	HTTP Method	파일명	로그인	페이지 설명
메인 화면			GET	main.html	X	메인 화면
로그인			POST	/member/login.html	X	로그인 화면
회원 가입			POST	/member/sign-up.html	X	회원 가입 화면
회원 수정			POST	/member/edit.html	O	회원 정보 수정 화면
회원 탈퇴			DELETE	/member/delete.html	O	회원 탈퇴 화면
회원 통계	남성		GET	/ideal/statistic-male.html	O	남성 회원 통계
	여성		GET	/ideal/statistic-female.html	O	여성 회원 통계
이상형 찾기	check		GET	/ideal/check-ideal.html	O	이상형 고르기 화면
	result		GET	/ideal/result.html	O	이상형 결과 화면
게시판	게시글 목록		GET	/board/list.html	O	게시글 목록 화면
	게시글 상세		GET	/board/board-detail.html	O	게시글 상세 화면
	게시글 작성		GET	/board/write	O	게시글 작성 화면
	게시글 수정		GET	/board/update	O	게시글 수정 화면

5. 사이트 구성

5.3 프로세스 정의



6. ERD



6. REST API

요청	URI
회원 가입 화면 / 회원 가입	(GET, POST) /member/sign-up
로그인 화면 / 로그인	(GET, POST) /member/login
로그아웃	POST /member/logout
이상형 선택 화면 / 이상형 선택	(GET, POST) /ideal/check
결과 화면	GET /ideal/result
게시글 목록	GET /board
게시글 상세	GET /board/{boardIdx}
게시글 작성 화면 / 게시글 작성	(GET, POST) /board/write
게시글 수정 화면 / 게시글 수정	(GET, POST) /board/update/{boardIdx}
게시글 삭제	DELETE /board/{boardIdx}
댓글 작성	POST /board/{boardIdx}

7. 핵심 기능

7.1 index

로그인하기

로그인 ID

비밀번호

main

홍길동님 환영합니다.

```
@GetMapping("/")  
public String home(HttpServletRequest request, Model model) {  
    HttpSession session = request.getSession(false);  
  
    if(session == null) {  
        return "/member/login";  
    }  
  
    Member member = (Member) session.getAttribute(SessionConst.sessionId);  
  
    model.addAttribute("member", member);  
    return "main";  
}
```

7. 핵심 기능

7.1 회원 가입

```
@Id @GeneratedValue(strategy = GenerationType.IDENTITY)
@Column(name = "idx")
private Long idx;

// member ID
@NotBlank(message = "ID는 필수 입력 항목입니다.")
@Column(name = "member_id")
private String memberId;

// member password
@NotBlank(message = "비밀번호는 필수 입력 항목입니다.")
@Column(name = "member_pw")
private String memberPw;

// member name
@NotBlank(message = "이름은 필수 입력 항목입니다.")
@Column(name = "member_name")
private String memberName;

// member age
@Enumerated(value = EnumType.STRING)
@Column(name = "member_age")
private MemberAge memberAge;

// gender
@Enumerated(value = EnumType.STRING)
@Column(name = "gender")
private Gender gender;

// member MBTI
@Enumerated(value = EnumType.STRING)
@Column(name = "mbti")
private Mbti mbti;

// member region
@Enumerated(value = EnumType.STRING)
@Column(name = "region")
private Region region;
```

Entity

View

ID :

비밀번호 :

이름 :

나이 :

성별 : ☐ 남성 ☐ 여성

MBTI :

거주지 :

Controller

```
// 회원가입
@PostMapping("/sign-up")
public String register(@Valid Member member,
                        BindingResult result,
                        Model model) {
    if (result.hasErrors()) {
        return "member/sign-up";
    }

    service.save(member);

    return "redirect:/";
}
```

7. 핵심 기능

7.1 회원 가입

```
// [회원가입]
@Override
public void save(Member member) {
    member.setMemberPw(bCryptPasswordEncoder.encode(member.getMemberPw()));
    memberRepository.save(member);
}
```

Service

idx	member_id	member_pw	member_name
1	aa	\$2a\$10\$YLVtaf3wgoe5T.vRaA5GtuaueG4i62MG...	홍길동
2	bb	\$2a\$10\$UctVEZ8GYQCTvMxkLbNRhOOMJM383...	카리나
3	cc	\$2a\$10\$ud8nb8f7dOvwf9aasDYTQutD5ekFyg2...	원터

DB(Table : member_info)

BCryptPasswordEncoder

Spring Security에서 지원하는 Class

.encode() method를 사용하여 비밀번호를 암호화 할 수 있다.

7. 핵심 기능

7.2 로그인

로그인하기

로그인 ID

비밀번호

View

```
@PostMapping("/login")
public String login(@Valid @ModelAttribute Member member,
                    BindingResult result,
                    HttpServletRequest request,
                    Model model) {
    // 입력된 ID/PW를 가진 회원이 있는지 확인
    Member loginMember = service.login(member.getMemberId(), member.getMemberPw());

    if(loginMember == null) {
        model.addAttribute("member", loginMember);
        return "redirect:login";
    }

    HttpSession session = request.getSession();
    session.setAttribute(SessionConst.sessionId, loginMember);

    Member userInfo = (Member) session.getAttribute(SessionConst.sessionId);
    model.addAttribute("member", userInfo);

    return "redirect:/";
}
```

Session에 member 객체 저장

Session에 저장된 member 객체를
참조변수 userInfo에 저장 → model을 통해 view로 전달

Controller

7. 핵심 기능

7.2 로그인

```
// [로그인]
@Override
public Member login(String memberId, String password) {
    Optional<Member> loginMember = memberRepository.findByMemberId(memberId);

    if(bCryptPasswordEncoder.matches(password, loginMember.get().getMemberPw())) {
        // 비밀번호가 일치할 경우
        return loginMember.get();
    }
    else { // 비밀번호가 불일치 할 경우
        return null;
    }
}
```

Service

BCryptPasswordEncoder

Spring Security에서 지원하는 Class

.matches() method를 사용하여 비밀번호를 복호화 할 수 있다.

main

홍길동님 환영합니다.

로그아웃

게시판

이상형 찾기

회원 통계

View

Model에서 보내준 userInfo의 .getMemberName()

7. 핵심 기능

7.3 게시판 – 게시판 목록

홍길동님이 로그인 중입니다.

순번	제목	작성자	작성 시간	수정 시간	조회수
1	aa	홍길동	2023-01-04T20:59:19	2023-01-04T20:59:19	14
2	bb	bb	2023-01-04T20:59:27	2023-01-04T20:59:27	3
3	aa	홍길동	2023-01-06T01:00:39	2023-01-06T01:00:39	0
4	cc	홍길동	2023-01-06T01:00:42	2023-01-06T01:00:42	0
5	a	홍길동	2023-01-06T01:00:44	2023-01-06T01:00:44	0
6	b	홍길동	2023-01-06T01:00:46	2023-01-06T01:00:46	0
7	2	홍길동	2023-01-06T01:00:48	2023-01-06T01:00:48	0
8	3	홍길동	2023-01-06T01:00:50	2023-01-06T01:00:50	0
9	1414	홍길동	2023-01-06T01:00:53	2023-01-06T01:00:53	0
10	aaf	홍길동	2023-01-06T01:00:57	2023-01-06T01:00:57	0

1 2

게시글 작성

메인으로

홍길동님이 로그인 중입니다.

순번	제목	작성자	작성 시간	수정 시간	조회수
11	afaf	홍길동	2023-01-06T01:01	2023-01-06T01:01	0
12	제목제목 😞	홍길동	2023-01-10T22:50:29	2023-01-10T22:50:29	1

1 2

게시글 작성

메인으로

```
// 게시판 목록
@GetMapping
public String boardList(@PageableDefault(page=0, size=10) Pageable pageable,
                        @RequestParam(value = "page", required = false) Integer page,
                        HttpServletRequest request,
                        Model model) {

    // 로그인/게시판 연동
    HttpSession session = request.getSession();
    Member member = (Member) session.getAttribute(SessionConst.SESSION_ID);

    Page<Board> list = boardService.findAllPage(pageable);

    // pageable은 0부터 시작 → 1을 처리하려면 +1
    int currentPage = list.getPageable().getPageNumber() + 1;

    // ex)현재 페이지 6 : 6페이지를 누르면 1페이지가 눈에 보이지 않게 -4
    int startPage = Math.max(currentPage - 4, 1);
    // ex)현재 페이지 1 : 1페이지에선 10페이지까지 눈에 보임
    int endPage = Math.min(currentPage + 9, list.getTotalPages());

    model.addAttribute("member", member);
    model.addAttribute("list", list);
    model.addAttribute("currentPage", currentPage);
    model.addAttribute("startPage", startPage);
    model.addAttribute("endPage", endPage);
    return "/board/list";
}
```

Spring Data Jpa에서 지원하는 Pageable Interface를 사용하여 간단하게 Paging 구현

7. 핵심 기능

7.3 게시판 - 게시물 상세

홍길동님이 로그인 중입니다.

순번 :

제목 :

내용 :

작성자 :

작성 시간 :

수정 시간 :

조회수 :

댓글을 남기세요

작성자	내용	작성 시간
홍길동	댓글댓글	2023-01-04T21:30
홍길동	댓글 내용내용	2023-01-04T21:36:45
조동현	악플악플	2023-01-04T21:39:27
조동현	수정된 내용~~	2023-01-04T21:56:56

작성자 :

댓글 :

게시글 상세 부분

게시글 마다 달 수 있는
댓글 부분

조회수 1 증가 Query

```
@GetMapping("/{boardIdx}")
public String board(@PathVariable(value="boardIdx", required = false) Long boardIdx,
                    @ModelAttribute Board board,
                    HttpServletRequest request,
                    Model model) {

    Board boardDetail = boardService.getBoard(board);

    // 조회수 1 증가
    boardService.hitCnt(boardIdx);

    log.info("getComment 실행 시작");

    // 댓글 불러오기
    List<Comment> commentList = boardService.getComment(boardIdx);

    log.info("getComment 실행 완료");

    HttpSession session = request.getSession();
    Member member = (Member)session.getAttribute(SessionConst.sessionId);

    log.info("member -> {}", member.getMemberName());

    model.addAttribute("member", member);
    model.addAttribute("boardDetail", boardDetail);
    model.addAttribute("comment", commentList);
    return "/board/board-detail";
}
```

게시글 상세
부분 불러오기

조회수 1 증가

댓글 목록 불러오기

게시판에서도 로그인 유지

```
public interface BoardRepository extends JpaRepository<Board, Long> {
    @Transactional
    @Modifying
    @Query("Update Board b Set b.hitCnt = b.hitCnt + 1 Where b.boardIdx = :boardIdx")
    public int updateHitCnt(@Param("boardIdx") Long boardIdx);
}
```

7. 핵심 기능

7.3 게시판 – 작성 시간 / 수정 시간

```
@MappedSuperclass
@EntityListeners(AuditingEntityListener.class)
public class BaseTime {

    @CreatedDate
    @Column(name = "create_date")
    private LocalDateTime createDate;

    @LastModifiedDate
    @Column(name = "update_date")
    private LocalDateTime updateDate;
}
```

```
@EnableJpaAuditing
public class ProjectApplication {

    public static void main(String[] args) {
        SpringApplication.run(ProjectApplication.class, args);
    }

}
```

```
@Entity
public class Board extends BaseTime {
```

```
@Entity
public class Comment extends BaseTime {
```

@MappedSuperclass

: 여러 테이블에서 공통적으로
매핑 정보가 필요할 때 사용.

@CreatedDate

: Entity가 생성되어 저장될 때 시간이 자동으로 저장

@LastModifiedDate

: Entity의 값을 변경할 때 시간이 자동 저장

7. 핵심 기능

7.3 게시판 - 게시글 작성

글 작성 화면

제목 :

작성자 :

내용 :

Controller

```
// insert
@GetMapping("/write")
public String insertBoard(HttpServletRequest request, Model model) {
    HttpSession session = request.getSession();
    Member member = (Member) session.getAttribute(SessionConst.sessionId);
    model.addAttribute("member", member);
    return "/board/write";
}

@PostMapping("/write")
public String write(Board board) {
    boardService.insertBoard(board);
    return "redirect:/board";
}
```

```
@Override
public void insertBoard(Board board) {
    boardRepository.save(board);
}
```

Service

board_idx	title	content	member_name	create_date	update_date	hit_cnt
1	aa	수정된 내용	홍길동	2023-01-04 20:59:19	2023-01-10 22:59:35	17
2	bb	bb	bb	2023-01-04 20:59:27	2023-01-04 20:59:27	4
3	aa	aa	홍길동	2023-01-06 01:00:39	2023-01-06 01:00:39	0
4	cc	cc	홍길동	2023-01-06 01:00:42	2023-01-06 01:00:42	0
5	a	a	홍길동	2023-01-06 01:00:44	2023-01-06 01:00:44	0
6	b	b	홍길동	2023-01-06 01:00:46	2023-01-06 01:00:46	0
7	2	2	홍길동	2023-01-06 01:00:48	2023-01-06 01:00:48	0
8	3	3	홍길동	2023-01-06 01:00:50	2023-01-06 01:00:50	0
9	1414	1441	홍길동	2023-01-06 01:00:53	2023-01-06 01:00:53	0
10	aaf	aaf	홍길동	2023-01-06 01:00:57	2023-01-06 01:00:57	0
11	afaf	aaf	홍길동	2023-01-06 01:01:00	2023-01-06 01:01:00	0
12	제...	내용 내용~...	홍길동	2023-01-10 22:50:29	2023-01-10 22:50:29	1

DB

7. 핵심 기능

7.3 게시판 - 게시물 수정

홍길동님이 로그인 중입니다.

순번 : 1
제목 : aa
내용 : aa
작성자 : 홍길동
작성 시간 : 2023-01-04T20:59:19
수정 시간 : 2023-01-04T20:59:19
조회수 : 14
[목록으로](#) [게시글 수정](#) [게시글 삭제](#)

순번 : 1
제목 : 수정된 제목
내용 : 수정된 내용
작성자 : 홍길동
작성 시간 : 23. 1. 4. 오후 4:53
수정 시간 : 23. 1. 4. 오후 4:53
조회수 : 22
[취소](#) [수정 완료](#)

Controller

```
@PostMapping("/update/{boardIdx}")  
public String update(@PathVariable(value="boardIdx", required=false) Long boardIdx,  
    @ModelAttribute Board board) {  
    boardService.updateBoard(board);  
    return "redirect:/board/" + boardIdx;  
}
```

```
@Override  
public void updateBoard(Board board) {  
    boardRepository.save(getBoard(board).update(board));  
}
```

해당 게시글만 수정 후 다시 저장

Service

홍길동님이 로그인 중입니다.

순번 : 1
제목 : 수정된 제목
내용 : 수정된 내용
작성자 : 홍길동
작성 시간 : 2023-01-04T16:53:59
수정 시간 : 2023-01-11T14:33:02
조회수 : 23
[목록으로](#) [게시글 수정](#) [게시글 삭제](#)

댓글을 남기세요

작성자	내용	작성 시간
홍길동	호호호호	2023-01-05T09:25:50
김길동	히히히히	2023-01-05T09:26:02
홍길동	히히히히gg	2023-01-05T11:38:26
김길동	fff	2023-01-05T11:38:49
홍길동	bb	2023-01-05T16:51:05

작성자 :
홍길동
댓글 :

[작성](#)

7. 핵심 기능

7.3 게시판 - 게시물 삭제

홍길동님이 로그인 중입니다.

순번	제목	작성자	작성 시간	수정 시간	조회수
11	브	홍길동	2023-01-11T15:52:08	2023-01-11T15:52:08	0
12	브	홍길동	2023-01-11T15:52:10	2023-01-11T15:52:10	1

[1](#) [2](#)

게시글 작성

메인으로

홍길동님이 로그인 중입니다.

순번 : 12

제목 : 브

내용 : 브

작성자 : 홍길동

작성 시간 : 2023-01-11T15:52:10

수정 시간 : 2023-01-11T15:52:10

조회수 : 1

목록으로

게시글 수정

게시글 삭제

```
// delete
@DeleteMapping("/delete/{boardIdx}")
public String delete(Board board) {
    boardService.deleteBoard(board);
    return "redirect:/board";
}
```

홍길동님이 로그인 중입니다.

순번	제목	작성자	작성 시간	수정 시간	조회수
11	브	홍길동	2023-01-11T15:52:08	2023-01-11T15:52:08	0

[1](#) [2](#)

게시글 작성

메인으로

7. 핵심 기능

7.3 게시판 - 댓글

홍길동님이 로그인 중입니다.

순번 :

제목 :

내용 :

작성자 :

작성 시간 :

수정 시간 :

조회수 :

댓글을 남기세요

작성자 내용 작성 시간
홍길동 댓글댓글 2023-01-11T17:07:20

작성자 :

댓글 :

```
// comment
@PostMapping("/{boardIdx}")
public String comment(@PathVariable(name = "boardIdx", required = false) Long boardIdx,
                      Comment comment) {
    boardService.insertComment(comment);
    return "redirect:/board/"+boardIdx;
}
```

해당 게시 글의 상세 페이지로 redirect 하기 때문에
작성 버튼을 클릭하자마자 댓글이 추가된 것이 보인다.

comment_idx	board_idx	content	member_name	create_date
7	1	악플악플	홍길동	2023-01-11 15:51:14
8	11	댓글댓글	홍길동	2023-01-11 17:07:20
9	11	댓글 댓글	홍길동	2023-01-11 17:07:48
10	11	악플악플	홍길동	2023-01-11 17:07:52

작성자 : Session에 저장된 member.getMemberName()
→ 로그인 되어있는 회원의 이름

7. 핵심 기능

7.4 이상형 찾기

```
@RequiredArgsConstructor
@Getter
public enum MemberAge {
    TWENTY("20대"), THIRTY("30대"),
    FOURTY("40대"), FIFTY("50대");

    private final String description;
}
```

```
@RequiredArgsConstructor
@Getter
public enum Region {
    SEOUL("서울"), INCHEON("인천"), DAEGU("대구"), GWANGJU("광주"),
    DAEJEON("대전"), ULSAN("울산"), BUSAN("부산"), JEJU("제주");

    private final String description;
}
```

```
public enum Mbti {
    INTJ, INTP, ENTJ, ENTP,
    INFJ, INFP, ENFJ, ENFP,
    ISTJ, ISFJ, ESTJ, ESFJ,
    ISTP, ISFP, ESTP, ESFP
}
```

연령대, 거주지, MBTI → ENUM으로 설정

View

연령대 : ☒ 20대 ☐ 30대 ☐ 40대 ☐ 50대
거주지 : ☒ 서울 ☐ 인천 ☐ 대구 ☐ 광주 ☐ 대전 ☐ 울산 ☐ 부산
MBTI : ☒ INTJ ☐ INTP ☐ ENTJ ☐ ENTP ☐ INFJ ☐ INFP ☐ ENFP



```
h.p.controller.ideal.IdealController : MemberAge → TWENTY
h.p.controller.ideal.IdealController : Region → SEOUL
h.p.controller.ideal.IdealController : MBTI → INTJ
```


7. 핵심 기능

7.4 이상형 찾기

```
@PostMapping("/check")
public String checkIdeal(@ModelAttribute Member member,
                        HttpServletRequest request,
                        RedirectAttributes redirectAttributes,
                        Model model) {
    HttpSession session = request.getSession();
    Member loginMember = (Member)session.getAttribute(SessionConst.sessionId);

    model.addAttribute("member", loginMember);

    log.info("member -> {}", loginMember);

    // 로그인 된 회원이 남성일 경우
    if(loginMember.getGender().equals(Gender.M)) {
        List<Member> femaleList = idealService.getIdealMale(member.getMemberAge(),
                                                            member.getRegion(),
                                                            member.getMbti());

        model.addAttribute("list", femaleList);
        return "ideal/result";
    }
    // 로그인 된 회원이 여성일 경우
    List<Member> maleList = idealService.getIdealFemale(member.getMemberAge(),
                                                        member.getRegion(),
                                                        member.getMbti());

    model.addAttribute("list", maleList);
    return "ideal/result";
}
```

BoardController

Controller에서 조건문을 사용하여

로그인 된 회원이 ‘남성’일 경우

→ 3가지 조건에 해당하는 여성만 검색

로그인 된 회원이 ‘여성’일 경우

→ 3가지 조건에 해당하는 남성만 검색

7. 핵심 기능

7.4 이상형 찾기

BoardRepository

```
final String strMale = "select m from Member m " +  
    "where m.memberAge like :memberAge " +  
    "and m.mbti like :mbti " +  
    "and m.region like :region " +  
    "and m.gender like 'F';" → 여성 회원만 검색  
  
@Query(strMale)  
public List<Member> getIdealMale(@Param("memberAge") MemberAge memberAge,  
    @Param("region") Region region,  
    @Param("mbti") Mbti mbti);  
  
final String strFemale = "select m from Member m " +  
    "where m.memberAge like :memberAge " +  
    "and m.mbti like :mbti " +  
    "and m.region like :region " +  
    "and m.gender like 'M';" → 남성 회원만 검색  
  
@Query(strFemale)  
public List<Member> getIdealFemale(@Param("memberAge") MemberAge memberAge,  
    @Param("region") Region region,  
    @Param("mbti") Mbti mbti);
```

result

홍길동님이 로그인 중입니다.



이름 나이 성별 MBTI 거주지
카리나 20대 여성 INTJ 서울
원터 20대 여성 INTJ 서울

다시 검색

메인으로

8. 트러블 슈팅

Trouble Shooting

🔥 문제 발생

조회수 증가 code 작성 후 실행

[Error 발생]

org.hibernate.hql.internal.QueryExecutionRequestException:

Not supported for DML operations [Update himedia.project.domain.board.Board b Set b.hitCnt = b.hitCnt + 1 Where b.boardIdx

🔥 문제 해결

updateHitCnt() method 위에 @Modifying Annotation을 추가해주었다.

DML(Insert, Update, Delete)을 사용할 때에는 반드시 @Modifying Annotation 사용할 것

```
public interface BoardRepository extends JpaRepository<Board, Long> {  
    @Transactional  
    @Modifying  
    @Query("Update Board b Set b.hitCnt = b.hitCnt + 1 Where b.boardIdx = :boardIdx")  
    public int updateHitCnt(@Param("boardIdx") Long boardIdx);  
}
```

🔥 문제 해결 후

조회수 : 52

순번	제목	좋아요	싫어요	작성 시간	수정 시간	조회수
1	수정된 안녕하세요	30	2	2022-12-29T20:55:28	2022-12-31T14:54:49	52
2	여보세요 📢	30	2	2022-12-29T20:55:29	2022-12-30T20:44:43	42
3	제목 제목 🗨️	0	0	2022-12-29T21:06:29	2022-12-29T21:06:29	3

게시글 작성

메인으로

조회수 : 53

순번	제목	좋아요	싫어요	작성 시간	수정 시간	조회수
1	수정된 안녕하세요	30	2	2022-12-29T20:55:28	2022-12-31T14:54:49	53
2	여보세요 📢	30	2	2022-12-29T20:55:29	2022-12-30T20:44:43	42
3	제목 제목 🗨️	0	0	2022-12-29T21:06:29	2022-12-29T21:06:29	3

게시글 작성

메인으로

9. 추후 작업

✓ Backend

- ✓ Member
 - 1. 회원가입 시 중복 체크 등 유효성 검사
 - 2. 회원가입/로그인 시 Error 메시지 출력
 - 3. My page 구현
- ✓ Board
 - 1. 자신이 작성한 글에 대해서만 수정/삭제가 가능하게 구현
 - 2. 작성/수정 시간이 좀 더 깔끔하게 출력되도록 구현
 - 3. 댓글 수정/삭제 기능 구현
- ✓ Ideal
 - 1. 체크박스 여러 개 중복 체크 할 수 있게 구현

✓ Frontend

- ✓ BootStrap 이용하여 View에 Style 추가

10. 소감

- ✓ 시간 분배의 중요성 !
- ✓ 개인 프로젝트라서 힘들고 버거운 정도 많았지만 처음부터 끝까지 혼자 할 수 있다는 장점 !

11. 출처

- <https://velog.io/@yeomyaloo/9.-%EC%8A%A4%ED%94%84%EB%A7%81%EB%B6%80%ED%8A%B8%EC%99%80-%ED%83%80%EC%9E%84%EB%A6%AC%ED%94%84%EB%A1%9C-%EA%B2%8C%EC%8B%9C%ED%8C%90-%EB%A7%8C%EB%93%A4%EA%B8%B0-%ED%8E%98%EC%9D%B4%EC%A7%95https://sanghye.tistory.com/36>
- <https://heewon26.tistory.com/125>
- <https://develop-writing.tistory.com/90>
- <https://velog.io/@won-developer/%EB%A1%9C%EA%B7%B8%EC%9D%B8-%EA%B5%AC%ED%98%84-%EC%BF%A0%ED%82%A4%EC%99%80-%EC%84%B8%EC%85%98-n3wwrtip#-%EC%BF%A0%ED%82%A4%EC%99%80-%EC%84%B8%EC%85%98>
- <https://1-7171771.tistory.com/78>
- <https://memory-develo.tistory.com/154>
- <https://jiurinie.tistory.com/70>
- <https://catsbi.oopy.io/0c27061c-204c-4fbf-acfd-418bdc855fd8>
- <https://velog.io/@codren/%ED%9A%8C%EC%9B%90-%EA%B0%80%EC%9E%85-%EA%B8%B0%EB%8A%A5-%EA%B5%AC%ED%98%84>
- <https://laycoder.tistory.com/106>
- <https://velog.io/@jln0616/%EC%8A%A4%ED%94%84%EB%A7%81-%EA%B2%8C%EC%8B%9C%ED%8C%90-%EC%9D%B4%EB%AF%B8%EC%A7%80-%EC%B2%A8%EB%B6%80>
- <https://blog.naver.com/PostView.naver?blogId=sosokhs79&logNo=20206455881&categoryNo=7&parentCategoryNo=0&viewDate=¤tPage=1&postListTopCurrentPage=1&from=postView>



들어 주셔서
감사합니다