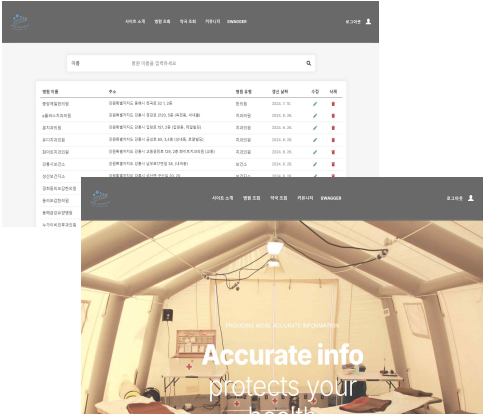

PORTFOLIO

Spring Boot와 AWS를 활용한 의료정보포탈 사이트 개발

About project

- Service Layer Architecture 사용
- Thymeleaf 템플릿 엔진을 사용하여 HTML을 동적으로 생성
- 사용자 인터페이스의 스타일링 및 반응형 디자인을 위해 Bootstrap 사용
- 공통 기능: 사이트 소개 확인, 병원 조회, 약국 조회
- 회원 기능: 사용자 프로필 관리, 병원 / 약국 북마크, 커뮤니티
- 관리자 기능: 회원 관리, 병원 / 약국 정보 최신화, 병원 / 약국 정보 수정 / 삭제, 커뮤니티 관리



Introduce project

작업 기간	2024. 04 ~ 2024. 07 (4개월)
인력 구성(기여도)	BE 1명 / FE 1명 (BE / FE 기여도 100%)
프로젝트 목적	개인 사이드 프로젝트로 Java와 Spring Boot 사용 경험 및 속도도 향상을 위하여 진행함
프로젝트 내용	사용자가 근처 병원 및 약국 정보를 빠르고 쉽게 조회하고 다른 사용자와 교류를 나눌 수 있는 커뮤니티 기능을 가진 의료정보포탈 사이트 개발
주요 업무 및 상세 역할	1) 병원 / 약국 조회 및 북마크 기능 구현 2) 사용자 / 관리자 프로필 관리 구현 3) 프로젝트 착수 단계부터 완료 단계까지의 진행 경험 4) Swagger 문서 작성 5) 사용자 / 관리자 매뉴얼 작성
사용언어 및 개발 환경	Java, Spring Boot, Thymeleaf, HTML5, CSS3, JavaScript, JQuery, Bootstrap, Eclipse, AWS
참고 자료	Github 링크: https://bit.ly/mediquanta 서비스 링크: www.mediquanta.com

Main work

프로젝트 설계

WBS

Version	Author	Description	Date			
0.1	안영욱	산출물리스트 WBS 1차 배포	2024-04-01			
0.2	권우진	산출물리스트 WBS 작성	2024-04-02			
Task ID	Task	Subtask	Start Date	End Date	Duration	Resource
학수	프로젝트 계획 수립	프로젝트 타임라인, 역할 및 책임, 리스크 관리 및 대응 전략	2024-04-02	2024-04-04	3 days	권우진
분석	요구사항 분석	요구사항 분석 회의	2024-04-05	2024-04-20	16 days	권우진
	시장 분석 및 조사	경쟁사 및 시스템 분석	2024-04-21	2024-05-01	11 days	권우진
설계	데이터베이스 테이블 설계	데이터베이스 테이블 및 키 정의	2024-05-02	2024-05-02	1 days	권우진
	프로세스 설계	프로세스 정의	2024-05-03	2024-05-06	4 days	권우진
	UI/UX 설계	UI/UX 정의	2024-05-06	2024-05-09	4 days	권우진
	엔드포인트 설계	엔드포인트 정의	2024-05-10	2024-05-11	2 days	권우진
	메뉴 설계	메뉴 구조 정의	2024-05-12	2024-05-13	2 days	권우진
	클래스 설계	클래스 정의	2024-05-14	2024-05-16	3 days	권우진
	기능 설계	기능 구성도 정의	2024-05-17	2024-05-19	3 days	권우진
	시스템 아키텍처 설계	CLI/CD 파이프라인 구축	2024-05-20	2024-05-27	8 days	권우진
	프록시선 환경 구축	AWS EC2 및 RDS 설정, Route53 도메인 구성	2024-05-28	2024-05-30	3 days	권우진
	테스트 설계	단위 테스트케이스 설계	2024-05-31	2024-06-02	3 days	권우진
	비즈니스 로직 개발	ModelAndView, REST API, Mybatis	2024-06-03	2024-06-15	13 days	권우진
	공통 모듈 개발	보통, 인증, 파일, 포털, 로거, AOP, 인터셉터, 필터	2024-06-16	2024-06-20	5 days	권우진
개발	Database	ORM 기능 개발	2024-06-21	2024-06-24	4 days	권우진
	프론트엔드 개발	사용자 친화적 UI 구현	2024-06-25	2024-06-30	6 days	권우진
	단위, 통합테스트	단위 테스트 및 통합 테스트 수행	2024-07-01	2024-07-03	3 days	권우진
구현	배포일 작성	관리자 및 사용자 메뉴일 작성	2024-07-04	2024-07-06	3 days	권우진
종료	프로젝트 완료	완료 보고서 작성 및 유지보수 계획 수립	2024-07-07	2024-07-08	2 days	권우진

프로젝트 계획서

강사	원장
	

프로젝트 계획서

2024-04-02

1. 프로젝트 개요

프로젝트명	Amazon Web Service 기반 의료 서비스 제공 사이트 개발 프로젝트
수행기간	2024 년 04 월 02 일 ~ 2024 년 07 월 08 일
장소	메가스터디 아이티 아카데미 (강남지점)
투입 공수	2M / M

Main work 프로젝트 분석

요구사항 정의서

요구사항 정의서

프로젝트	amazon Web Service 기반 의료 정보 포털 시스템 개발 프로젝트				작성자	권우진
시스템명	Mediquanta				작성일	2024-04-09
요구사항	업무구분	요구사항	유형	처리방안		
RQ-001	사용자	회원가입	기능	스프링 MVC 개발		
RQ-002	사용자	로그인	기능	스프링 MVC 개발		
RQ-003	사용자	회원 정보 수정	기능	스프링 MVC 개발		
RQ-004	사용자	커뮤니티 댓글 등록	기능	스프링 MVC 개발		
RQ-005	사용자	댓글 삭제	기능	스프링 MVC 개발		
RQ-006	사용자	커뮤니티 댓글 등록	기능	스프링 MVC 개발		
RQ-007	사용자	약국 검색	기능	스프링 MVC 개발		
RQ-008	사용자	병원 검색	기능	스프링 MVC 개발		
RQ-009	사용자	약국 검색	기능	스프링 MVC 개발		
RQ-010	사용자	병원 위치 지도 표시	기능	스프링 MVC 개발		
RQ-011	사용자	약국 위치 지도 표시	기능	스프링 MVC 개발		
RQ-012	사용자	사용자 프로필 조회	기능	스프링 MVC 개발		
RQ-013	사용자	사용자 프로필 수정	기능	스프링 MVC 개발		
RQ-014	사용자	커뮤니티 게시물 등록	기능	스프링 MVC 개발		
RQ-015	사용자	커뮤니티 게시물 작성	기능	스프링 MVC 개발		
RQ-016	사용자	커뮤니티 게시물 수정	기능	스프링 MVC 개발		
RQ-017	사용자	본인이 작성한 게시물에 대해서 커뮤니티 게시물 삭제	기능	스프링 MVC 개발		
RQ-018	사용자	커뮤니티 댓글 작성	기능	스프링 MVC 개발		
RQ-019	사용자	커뮤니티 댓글 수정	기능	스프링 MVC 개발		
RQ-020	사용자	본인이 작성한 댓글에 대해서 커뮤니티 댓글 삭제	기능	스프링 MVC 개발		
RQ-021	사용자	댓글 북마크	기능	스프링 MVC 개발		
RQ-022	사용자	약국 북마크	기능	스프링 MVC 개발		
RQ-023	사용자	사용자 프로필에서 본인의 주소에서 가까운 병원 확인	기능	스프링 MVC 개발		
RQ-024	사용자	사용자 프로필에서 본인의 주소에서 가까운 약국 확인	기능	스프링 MVC 개발		
RQ-025	사용자	회원 탈퇴	기능	스프링 MVC 개발		
RQ-026	관리자	관리자 프로필 조회	기능	스프링 MVC 개발		
RQ-027	관리자	회원 검색 목록 조회	기능	스프링 MVC 개발		
RQ-028	관리자	커뮤니티 글인 (글인원 커뮤니티만 목록에 표시)	기능	스프링 MVC 개발		
RQ-029	관리자	병원 정보 추가	기능	스프링 MVC 개발		
RQ-030	관리자	병원 정보 수정	기능	스프링 MVC 개발		
RQ-031	관리자	병원 정보 삭제	기능	스프링 MVC 개발		
RQ-032	관리자	약국 정보 추가	기능	스프링 MVC 개발		
RQ-033	관리자	약국 정보 수정	기능	스프링 MVC 개발		
RQ-034	관리자	약국 정보 삭제	기능	스프링 MVC 개발		
RQ-035	관리자	병원 현황 확인	기능	스프링 MVC 개발		
RQ-036	관리자	약국 현황 확인	기능	스프링 MVC 개발		
RQ-037	관리자	디자인 레이아웃	디자인	Thymeleaf 활용		
RQ-038	관리자	디자인 레이아웃	디자인	Real Bootstrap 5 template 활용		

현행 시장 조사

약국찾기 - 전국 약국 정보

<https://pharm.goodhoorank.com/index.php>

일반	사이트 소개	사이트 소개
조회	약국 조회	전체 약국 목록
		평일 영업 약국
		평일 야간 영업 약국
		토요일 영업 약국
		토요일 야간 영업 약국
		일요일 영업 약국
		일요일 야간 영업 약국
		공휴일 영업 약국
		공휴일 야간 영업 약국
		오늘의 추천 약국
메디네비	http://medinavi.co.kr/	약국 검색
		지도별 약국 확인
		약국 후기
		최근 검색 지역
		최근 검색 약국
		최근 개원한 약국
		-

검색	병원 검색	-
병원 조회	지역별	지역별 병원 검색
	종류별	종류별 병원 검색
	진료과목별	진료과목별 병원 검색
	특수진료	특수진료 병원 검색
	명절 진료	명절 진료 병원 검색
의료 정보 조회	의료수가	의료수가 정보 조회
	질병 통계	질병 통계 조회
	약국	요일별 약국 조회
	의약품	의약품 조회
	의약품	의약품 조회
	약품식별	약품식별 정보 조회
	약품색	약품색 조회
메디사전	건강FAQ	건강 관련 FAQ
	식품	식품, 건강 관련 정보 제공

Main work ① 회원가입

- 우편번호 칸을 클릭하면 postcode.v2.js 실행되며 주소 클릭 시 지번 주소, 도로명 주소 칸이 자동으로 입력됨
- 입력폼을 전부 입력해야만 회원가입 가능

회원가입

아이디* 중복 확인 닉네임* 중복 확인

사용할 아이디를 입력해주세요. 사용할 닉네임을 입력해주세요.

비밀번호*

영어 소문자, 숫자, 특수문자를 포함한 10자리 이상

비밀번호 재확인*

비밀번호를 다시 입력해주세요.

프로필*

이메일*

you@example.com

우편번호*

해당 칸을 클릭하여 우편번호를 입력해주세요.

도로명 주소

우편번호 선택 시 자동 입력됩니다

지번 주소

우편번호 선택 시 자동 입력됩니다

기타 주소

기타 주소를 입력해주세요

성별* 성별*

연도-월-일 남

☐ 개인정보 수집 및 이용에 동의합니다.*

```

@GetMapping("/register")
@Operation(summary="페이지 이동 - 회원 등록", description="회원 등록 페이지로 이동")
public String register() {
    return "member/register";
}

@PostMapping("/validId")
@ResponseBody
@Operation(summary="아이디 확인", description="아이디를 입력받아 DB에 아이디가 존재하는지 확인")
public String validId(@RequestParam("memberId") String memberId) {
    return memberService.checkValidId(memberId);
}

@PostMapping("/validNickname")
@ResponseBody
@Operation(summary="닉네임 확인", description="닉네임을 입력받아 DB에 닉네임이 존재하는지 확인")
public String validNickname(@RequestParam("nickname") String nickname) {
    return memberService.checkValidNickname(nickname);
}

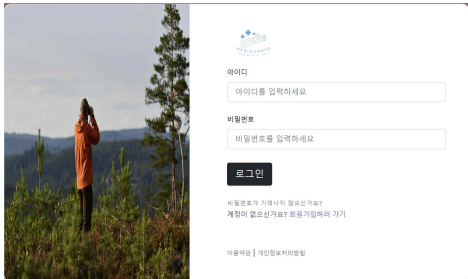
@PostMapping("/validEmail")
@ResponseBody
@Operation(summary="이메일 확인", description="이메일을 입력받아 DB에 이메일이 존재하는지 확인")
public String validEmail(@RequestParam("email") String email) {
    return memberService.checkValidEmail(email);
}

@PostMapping("/register")
@Operation(summary="회원 등록", description="사용자가 입력받은 정보를 토대로 회원 등록")
public String register(@RequestParam("uploadProfile") MultipartFile uploadProfile, @ModelAttribute MemberDTO memberDTO) throws IllegalStateException, IOException {
    memberService.createMember(uploadProfile, memberDTO);
    return "member/login";
}

```

Main work ② 로그인

- 아이디와 비밀번호를 입력받아 로그인
- 비밀번호를 잊어버렸을 경우 회원가입 시 입력한 이메일로 임시 비밀번호 전송



```
@PostMapping("/checkRole")
@ResponseBody
@Operation(summary="권한 확인", description="아이디를 입력받아 권한이 있는지 확인")
public String checkRole(@RequestParam("memberId") String memberId) {
    return memberService.checkRole(memberId);
}

@GetMapping("/login")
@Operation(summary="페이지 이동 - 로그인", description="로그인 페이지로 이동")
public String login() {
    return "member/login";
}

@PostMapping("/login")
@ResponseBody
@Operation(summary="로그인", description="아이디와 비밀번호를 입력받아 로그인")
public String login(@RequestBody MemberDTO memberDTO, HttpServletRequest request) {
    String isValidMember = "n";
    if (memberService.login(memberDTO)) {
        HttpSession session = request.getSession();
        session.setAttribute("memberId", memberDTO.getMemberId());
        session.setAttribute("role", checkRole(memberDTO.getMemberId()));
        isValidMember = "y";
    }
    return isValidMember;
}

@GetMapping("/forgotPasswd")
@Operation(summary="페이지 이동 - 비밀번호 찾기", description="비밀번호 찾기 페이지로 이동")
public String tempPasswd() {
    return "member/forgotPasswdForm";
}
```

Main work ③ 임시 비밀번호 발급

- 비밀번호를 잊어버렸을 경우 로그인 페이지에서 '비밀번호를 잊어버리셨나요?'를 클릭 후 임시 비밀번호 발급 페이지로 이동
- 회원가입 시 입력했던 이메일 주소를 입력하면 해당 이메일 주소가 있는지 확인하고 있다면 해당 이메일로 임시 비밀번호 발급



비밀번호를 잊어버리셨나요?

제 가지 귀운 단계로 비밀번호를 변경하세요.
이렇게 하면 새 비밀번호가 안전하게 유지됩니다.

1. 아래에 회원가입 시 생성하신 회원 아이디를 입력해주세요.
2. 입력하신 회원 아이디가 존재하는 경우
이메일로 임시 비밀번호를 발송합니다.
입력하신 회원 아이디가 존재하지 않는 경우
회원가입 화면으로 이동합니다.
3. 임시 비밀번호로 로그인 후 회원 수정에서 비밀번호를
변경해주세요.

아이디

임시 비밀번호 발급

[Mediquanta] 임시 비밀번호 발급

denw@denw@gmail.com
안녕하세요. 반갑습니다. 귀국 조희 플랫폼 Mediquanta입니다. 임시 비밀번호를 발송해드립니다. 아래 발송된 비밀번호로 로그인해주세요. 임시 비밀번호입니다.

4월 21일 (일) 오후 5:04
denw@denw@gmail.com
1/1

안녕하세요.

편리한 병원 및 약국 조회 플랫폼 Mediquanta입니다.

임시 비밀번호를 발송드립니다. 아래 발송된 비밀번호로 로그인해주세요.

```


@GetMapping("/forgotPasswd")
@Operation(summary="페이지 이동 - 비밀번호 찾기", description="비밀번호 찾기 페이지로 이동")
public String tempPasswd() {
    return "member/forgotPasswdForm";
}

@PostMapping("/forgotPasswd")
@Operation(summary="비밀번호 찾기", description="아이디를 통해 DB에 회원 정보가 있다면 임시 비밀번호를 회원의 이메일로 발송")
public String forgotPasswd(@RequestParam("memberId") String memberId) {
    MemberDTO memberDTO = memberService.getUserInfo(memberId);
    String page = "";
    if (memberDTO == null) {
        page = "/member/register";
    } else {
        String email = memberDTO.getEmail();
        EmailMessage emailMessage = EmailMessage.builder()
            .to(email)
            .subject("[Mediquanta] 임시 비밀번호 발급")
            .build();
        emailService.sendMail(emailMessage, "email");
        page = "/member/login";
    }
    return "redirect:" + page;
}

```


Main work ④ 사용자 프로필

메인 / 사용자 프로필



it's

북마크 회원 정보 수정 회원 탈퇴

닉네임	it's
이메일	you@gmail.com
회원번호	49405
도로명 주소	부산 사하구 괴정로 134 (괴정동)
지번 주소	부산 사하구 괴정동 1234
기타 주소	성광로조영아파트 101동

근처 병원 현황

온치과의원
 병원 타입: 치과의원
 주소: 부산광역시 사하구 낙동대로 428, 5층 (당리동)
 전화번호: 051-208-2275

박가정의학과의원
 병원 타입: 의원
 주소: 부산광역시 사하구 제석로 68, (당리동)
 전화번호: 051-747-2283

장사랑내과영상상의학과의원
 병원 타입: 의원
 주소: 부산광역시 사하구 낙동대로 438, 2,3,5층 (당리동, 한국해디칼)
 전화번호: 051-207-1088

근처 약국 현황

동해약국
 주소: 부산광역시 사하구 낙동대로 363, 1층 (당리동)
 전화번호: 051-203-7837

가온약국
 주소: 부산광역시 사하구 동해로 167, 101호 (당리동)
 전화번호: 051-996-4664

당리수정약국
 주소: 부산광역시 사하구 다대로 3, (당리동)
 전화번호: 051-203-6636


- 메인 화면에서 우측 상단의 프로필 아이콘을 클릭하여 프로필을 확인합니다
- 근처 병원 및 약국은 회원 정보의 주소를 기준으로 3킬로미터 이내에 위치한 병원/약국들을 찾고, 그들 중 가장 가까운 병원/약국 3개를 보여줍니다
- 프로필 화면에서 회원 정보 수정 및 회원 탈퇴가 가능하며 회원 탈퇴 시 회원 정보가 비활성화됩니다

```

<!-- 근처 병원 조회하기 -->
<select id="selectNearestHospitals" parameterType="map" resultType="HospitalDTO">
  <![CDATA[
    SELECT *
    FROM (
      SELECT
        HOSPITAL.*,
        6371 * ACOS(
          COS(RADIANS(#latitude))) *
          COS(RADIANS(HOSPITAL.LATITUDE)) *
          COS(RADIANS(HOSPITAL.LONGITUDE) - RADIANS(#longitude)) +
          SIN(RADIANS(#latitude))) *
          SIN(RADIANS(HOSPITAL.LATITUDE))
        ) AS DISTANCE
      FROM HOSPITAL
    ) DATA
    WHERE DATA.DISTANCE < 3
    ORDER BY DATA.DISTANCE
    LIMIT 3;
  ]]>
</select>
  
```

Main work ⑤ 관리자 프로필

관리자 / 관리자 프로필



관리자

회원 정보 수정

회원 관리

병원 현황

약국 현황

커뮤니티 현황

닉네임 관리자

이메일

우편번호

도로명 주소

지번 주소

가다 주소

현재 등록된 병원 현황

시/도	개수
서울	2381
경기	1516
부산	460
대구	326
인천	293
경남	253
대전	210
광주	175
충남	166
경북	165
전북	147
충북	124
울산	122
강원	109
전남	102
제주	41
세종시	16

현재 등록된 약국 현황

시/도	개수
경기	4316
서울	4058
부산	1284
경남	1033
대구	955
인천	834
경북	878
전북	780
충남	740
전남	646
대전	593
광주	579
강원	567
충북	556
울산	332
제주	248
세종시	122

- 관리자 프로필에서 회원 정보 수정 및 회원 · 커뮤니티를 관리할 수 있습니다
- 현재 등록된 병원 및 약국 데이터 현황이 지역을 기준으로 표시됩니다
- 불러오기 버튼 클릭 시 해당 항목의 데이터가 최신화됩니다

```

@Override
public void saveHospitalsToDatabase() {
    try {
        for (SgguCd sgguCd : SgguCd.values()) {
            int currentPage = 1;
            boolean hasNextPage = true;

            while (hasNextPage) {
                List<HospitalApiDTO> hospitals = hospitalApiManager.fetchByCode(currentPage, DEFAULT_NUM_OF_ROWS, sgguCd);

                if (hospitals.isEmpty()) {
                    hasNextPage = false;
                    continue;
                }

                for (HospitalApiDTO hospital : hospitals) {
                    HospitalDTO hospitalDTO = new HospitalDTO();
                    hospitalDTO.setHospitalName(hospital.getHospitalName());
                    hospitalDTO.setType(hospital.getType());
                    hospitalDTO.setSidoCdNm(hospital.getSidoCdNm());
                    hospitalDTO.setSgguCdNm(hospital.getSgguCdNm());
                    hospitalDTO.setAddress(hospital.getAddress());
                    hospitalDTO.setPhone(hospital.getPhone());
                    hospitalDTO.setLatitude(hospital.getLatitude());
                    hospitalDTO.setLongitude(hospital.getLongitude());
                    hospitalDTO.setHospitalUrl(hospital.getHospitalUrl());
                    hospitalDTO.setCreatedAt(new Date());
                    hospitalDTO.setUpdatedAt(null);

                    hospitalDAO.saveHospital(hospitalDTO);

                    dataCnt++;
                }


                log.info(sgguCd.toString() + " 지역코드 " + currentPage + " 페이지의 병원 데이터 저장");
                log.info("현재 저장된 데이터 개수: " + dataCnt + "개");

                currentPage++;
            }
        }
    } catch (ParseException e) {
        log.warn("병원 데이터 가져오기 실패: " + e.getMessage());
    }
}

```

Main work ④ 회원 정보 수정

백민 / 사용자 프로필 / 회원 정보 수정



it's

아이디	itsmo
비밀번호	새로운 비밀번호를 입력해주세요
프로필	파일 선택 선택된 파일 없음
이메일	you@gmail.com 등록 확인
닉네임	it's 등록 확인
성별	남 ▼
생일	1995-07-05 📅
우편번호	49406
도로명 주소	부산 사하구 괴정로 134 (괴정동)
지번 주소	부산 사하구 괴정동 1234
가타 주소	성경로즈힐아파트 101동

회원 정보 저장

- 프로필 화면에서 회원 정보 수정 버튼을 클릭하여 회원 정보를 수정할 수 있습니다.
- 입력 폼 안의 모든 항목을 입력해야 회원 정보 수정이 가능합니다.
- 우편번호 입력 칸 클릭 시 daum postcode 창이 뜨며 우편번호를 통해 주소를 검색할 수 있습니다
- 주소 선택 시 도로명 주소와 지번 주소는 자동으로 입력됩니다.

```
function execDaumPostcode() {
    new daum.Postcode({
        oncomplete: function(data) {
            var fullRoadAddr = data.roadAddress;
            var extraRoadAddr = '';

            if (data.bname !== '' && /[동|로|가]$/g.test(data.bname)){
                extraRoadAddr += data.bname;
            }

            if (data.buildingName !== '' && data.apartment === 'Y'){
                extraRoadAddr += (extraRoadAddr !== '' ? ', ' + data.buildingName : data.buildingName);
            }

            if (extraRoadAddr !== ''){
                extraRoadAddr = ' (' + extraRoadAddr + ')';
            }
            if (fullRoadAddr !== ''){
                fullRoadAddr += extraRoadAddr;
            }

            document.getElementById('zipcode').value = data.zonecode;
            document.getElementById('roadAddress').value = fullRoadAddr ? fullRoadAddr : "-";
            document.getElementById('landAddress').value = data.jibunAddress ? data.jibunAddress : "-";

        }
    }).open();
}
```

Main work ⑥ 병원 조회

- 공통 기능: 검색을 통해 이름 / 지역 / 유형별로 병원 검색
- 사용자: 병원 정보 북마크
- 관리자: 병원 정보 수정 & 삭제

사용자

이름

병원 이름을 입력하세요

Q

병원 이름	주소	병원 유형	경신 날짜	북마크
중앙재활의원	강원특별자치도 동해시 한국로 52, (전국동)	한의원	2024. 6. 26.	☆
e플러스치과의원	강원특별자치도 강릉시 경강로 2120, 5층 (옥천동, 서내동)	치과의원	2024. 6. 26.	☆

관리자

이름

병원 이름을 입력하세요

Q

병원 이름	주소	병원 유형	경신 날짜	수정	삭제
중앙재활의원	강원특별자치도 동해시 한국로 52 1, 2층	한의원	2024. 7. 10.		
e플러스치과의원	강원특별자치도 강릉시 경강로 2120, 5층 (옥천동, 서내동)	치과의원	2024. 6. 26.		

```

@ApiResponses(value = {
    @ApiResponse(responseCode = "200",
        description = "병원 조회 페이지 이동 성공",
        content = @Content(
            mediaType = "text/html",
            schema = @Schema(
                type = "string",
                example = "<html><body><div>병원 조회 페이지로 성공적으로 이동했습니다.</div></body></html>"
            )
        )
    },
    @ApiResponse(
        responseCode = "404",
        description = "페이지를 찾을 수 없음",
        content = @Content(
            schema = @Schema(
                implementation = NotFoundError.class
            ),
            examples = @ExampleObject(
                name = "Not Found",
                value = "{ \"message\": \"요청한 리소스를 찾을 수 없습니다.\", \"errorCode\": \"404\" }"
            )
        )
    )
})

@Operation(summary="페이지 이동 - 병원 조회", description="병원 조회 페이지로 이동")
@GetMapping
public String viewHospitalList(Model model, HttpSession session) {
    model.addAttribute("role", (String)session.getAttribute("role"));
    return "hospital/hospitalList";
}

```

```

@ApiResponses(value = {
    @ApiResponse(responseCode = "200",
        description = "병원 조회 성공",
        content = @Content(
            mediaType = "array",
            schema = @Schema(
                implementation=HospitalDTO.class
            )
        )
    },
    @ApiResponse(
        responseCode = "400",
        description = "잘못된 접근",
        content = @Content(
            schema = @Schema(
                implementation = BadRequestError.class
            ),
            examples = @ExampleObject(
                name = "Bad Request",
                value = "{ \"message\": \"잘못된 요청입니다.\", \"errorCode\": \"400\" }"
            )
        )
    )
})

@Operation(summary="병원 조회", description="병원 목록 가져오기")
@PostMapping("/getHospitalList")
@ResponseBody
public List<HospitalDTO> getHospitalList() {
    return hospitalService.getHospitalList();
}

```

Main work ⑥ 병원 조회

이름으로 병원 검색

```

@ApiResponses(value = {
    @ApiResponse(responseCode = "200",
        description = "병원 조회 성공",
        content = @Content(
            schema = @Schema(
                implementation = HospitalDTO.class
            )
        )
    ),
    @ApiResponse(
        responseCode = "400",
        description = "잘못된 접근",
        content = @Content(
            schema = @Schema(
                implementation = BadRequestError.class
            ),
            examples = @ExampleObject(
                name = "Bad Request",
                value = "{ \"message\": \"잘못된 요청입니다\", \"errorCode\": \"400\" }"
            )
        )
    )
})
@Operation(summary = "병원 이름으로 병원 조회", description = "병원 이름으로 병원 조회")
@PostMapping("/searchHospitalByName")
@ResponseBody
public List<HospitalDTO> searchHospitalByName(@RequestBody SearchData searchData) {
    return hospitalService.searchHospitalByName(searchData.getName());
}

```

시도 명으로 병원 검색

```

@ApiResponses(value = {
    @ApiResponse(responseCode = "200",
        description = "병원 조회 성공",
        content = @Content(
            mediaType = "array",
            schema = @Schema(
                implementation = HospitalDTO.class
            )
        )
    ),
    @ApiResponse(
        responseCode = "400",
        description = "잘못된 접근",
        content = @Content(
            schema = @Schema(
                implementation = BadRequestError.class
            ),
            examples = @ExampleObject(
                name = "Bad Request",
                value = "{ \"message\": \"잘못된 요청입니다\", \"errorCode\": \"400\" }"
            )
        )
    )
})
@Operation(summary = "시도 코드 명으로 병원 조회", description = "시도 코드 명으로 병원 조회")
@PostMapping("/searchHospitalBySidoCdNm")
@ResponseBody
public List<HospitalDTO> searchHospitalBySidoCdNm(@RequestBody SearchData searchData) {
    return hospitalService.searchHospitalBySidoCdNm(searchData.getSidoCdNm());
}

```

유형으로 병원 검색

```

@ApiResponses(value = {
    @ApiResponse(responseCode = "200",
        description = "병원 조회 성공",
        content = @Content(
            mediaType = "array",
            schema = @Schema(
                implementation = HospitalDTO.class
            )
        )
    ),
    @ApiResponse(
        responseCode = "400",
        description = "잘못된 접근",
        content = @Content(
            schema = @Schema(
                implementation = BadRequestError.class
            ),
            examples = @ExampleObject(
                name = "Bad Request",
                value = "{ \"message\": \"잘못된 요청입니다\", \"errorCode\": \"400\" }"
            )
        )
    )
})
@Operation(summary = "병원 유형으로 병원 조회", description = "병원 유형으로 병원 조회")
@PostMapping("/searchHospitalByType")
@ResponseBody
public List<HospitalDTO> searchHospitalByType(@RequestBody SearchData searchData) {
    return hospitalService.searchHospitalByType(searchData.getType());
}

```

Main work ⑦ 병원 수정

- 관리자만 가능하며 병원 수정 아이콘 클릭 시 수정 페이지로 이동합니다
- 병원 수정 시 위도, 경도 조회 기능은 카카오 API를 사용하였습니다

메인 / 병원 목록 / 중앙제일한의원

중앙제일한의원

병원 유형

한의원

주소 (시, 도)

강원

주소 (구, 동)

동해시

주소 (전체 주소)

강원특별자치도 동해시 천곡로 52 1, 2층

전화번호

033-535-1136

위도

37.5237799

경도

129.1109039

홈페이지

http://www.jhaniwon.com

주소

수정

```
@Operation(summary="페이지 이동 - 병원 수정", description="병원 수정 화면으로 이동")
@GetMapping("/viewHospitalUpdate")
public String viewHospitalUpdate(@RequestParam("hospitalId") long hospitalId, Model model) {
    HospitalDTO hospitalDTO = hospitalService.getHospitalDetails(hospitalId);
    model.addAttribute("hospital", hospitalDTO);
    return "hospital/hospitalUpdate";
}
```

```
@Operation(summary="병원 위치 조회", description="병원 전체 주소를 통해 병원 위치(위도, 경도) 조회")
@GetMapping("/findLoc")
@ResponseBody
public Map<String, Double> getLatitudeLongitude(@RequestParam("address") String address) {
    return hospitalService.kakaoLocalAPI(address);
}
```

```
@Operation(summary="병원 정보 수정", description="병원 정보 수정")
@PostMapping("/updateHospInfo")
public String updateHospInfo(@ModelAttribute HospitalDTO hospitalDTO) {
    hospitalService.updateHospInfo(hospitalDTO);
    return "redirect:/hospital";
}
```

Main work ⑧ 약국 조회

- 공통 기능: 검색을 통해 이름 / 지역별로 약국 검색
- 사용자: 약국 정보 북마크
- 관리자: 약국 정보 수정 & 삭제

사용자

약국 이름	주소	경선 날짜	북마크
프라자건강약국	강원특별자치도 원주시 농라동길 70, (무실동)	2024. 6. 28.	☆
신흥약국	강원특별자치도 원주시 원일로 91, (일산동)	2024. 6. 28.	☆

관리자

약국 이름	주소	경선 날짜	수정	삭제
프라자건강약국	강원특별자치도 원주시 농라동길 70, (무실동)	2024. 6. 28.		
신흥약국	강원특별자치도 원주시 원일로 91, (일산동)	2024. 6. 28.		

```

@ApiResponses(value = {
    @ApiResponse(responseCode = "200",
        description = "약국 조회 페이지 이동 성공",
        content = @Content(
            mediaType = "text/html",
            schema = @Schema(
                type = "string",
                example = "<html><body>약국 조회 페이지로 성공적으로 이동했습니다.</body></html>"
            )
        )
    },
    @ApiResponse(
        responseCode = "404",
        description = "페이지를 찾을 수 없음",
        content = @Content(
            schema = @Schema(
                implementation = NotFoundError.class
            ),
            examples = @ExampleObject(
                name = "Not Found",
                value = "{ \"message\": \"요청한 리소스를 찾을 수 없습니다.\", \"errorCode\": \"404\" }"
            )
        )
    )
})
@Operation(summary="페이지 이동 - 약국 조회", description="약국 조회 페이지로 이동")
@GetMapping
public String viewPharmacyList(Model model, HttpSession session) {
    model.addAttribute("role", (String) session.getAttribute("role"));
    return "pharmacy/pharmacyList";
}

```

```

@ApiResponses(value = {
    @ApiResponse(responseCode = "200",
        description = "약국 조회 성공",
        content = @Content(
            mediaType = "array",
            schema = @Schema(
                implementation=PharmacyDTO.class
            )
        )
    },
    @ApiResponse(
        responseCode = "400",
        description = "잘못된 접근",
        content = @Content(
            schema = @Schema(
                implementation = BadRequestError.class
            ),
            examples = @ExampleObject(
                name = "Bad Request",
                value = "{ \"message\": \"잘못된 요청입니다.\", \"errorCode\": \"400\" }"
            )
        )
    )
})
@Operation(summary="약국 조회", description="약국 목록 가져오기")
@PostMapping("/getPharmacyList")
@ResponseBody
public List<PharmacyDTO> getPharmacyList() {
    return pharmacyService.getPharmacyList();
}

```

Main work ⑧ 약국 조회

이름으로 병원 검색

```
@ApiResponses(value= {
    @ApiResponse(responseCode = "200",
        description = "약국 조회 성공",
        content = @Content(
            schema = @Schema(
                implementation=PharmacyDTO.class
            )
        )
    ),
    @ApiResponse(
        responseCode = "400",
        description = "잘못된 접근",
        content = @Content(
            schema = @Schema(
                implementation = BadRequestError.class
            ),
            examples = @ExampleObject(
                name = "Bad Request",
                value = "{ \"message\": \"잘못된 요청입니다\", \"errorCode\": \"400\" }"
            )
        )
    )
})
@Operation(summary="약국 이름으로 약국 조회", description="약국 이름으로 약국 조회")
@PostMapping("/searchPharmacyByName")
@ResponseBody
public List<PharmacyDTO> searchPharmacyByName(@RequestBody SearchData searchData) {
    return pharmacyService.searchPharmacyByName(searchData.getName());
}
```

시도 명으로 병원 검색

```
@ApiResponses(value= {
    @ApiResponse(responseCode = "200",
        description = "약국 조회 성공",
        content = @Content(
            mediaType = "array",
            schema = @Schema(
                implementation=PharmacyDTO.class
            )
        )
    ),
    @ApiResponse(
        responseCode = "400",
        description = "잘못된 접근",
        content = @Content(
            schema = @Schema(
                implementation = BadRequestError.class
            ),
            examples = @ExampleObject(
                name = "Bad Request",
                value = "{ \"message\": \"잘못된 요청입니다\", \"errorCode\": \"400\" }"
            )
        )
    )
})
@Operation(summary="시도 코드 명으로 약국 조회", description="시도 코드 명으로 약국 조회")
@PostMapping("/searchPharmacyBySidoCdNm")
@ResponseBody
public List<PharmacyDTO> searchPharmacyBySidoCdNm(@RequestBody SearchData searchData) {
    return pharmacyService.searchPharmacyBySidoCdNm(searchData.getSidoCdNm());
}
```


Main work ⑨ 약국 수정

- 관리자만 가능하며 약국 수정 아이콘 클릭 시 수정 페이지로 이동합니다
- 약국 수정 시 위도, 경도 조회 기능은 카카오 API를 사용하였습니다

건강과행복이열리는중앙메디칼약국

주소 (시, 도)

서울

주소 (구, 동)

동작구

주소 (전체 주소)

서울 동작구 흑석로 106-5 1층

전화번호

02-811-1700

위도

위도 / 경도 조회

37.5078916

경도

126.9614593

취소

수정

```
@Operation(summary="페이지 이동 - 약국 수정", description="약국 수정 화면으로 이동")
@GetMapping("/viewPharmacyUpdate")
public String viewPharmacyUpdate(@RequestParam("pharmacyId") long pharmacyId, Model model) {
    PharmacyDTO pharmacyDTO = pharmacyService.getPharmacyDetails(pharmacyId);
    model.addAttribute("pharmacy", pharmacyDTO);
    return "pharmacy/pharmacyUpdate";
}
```

```
@Operation(summary="약국 위치 조회", description="약국 전체 주소를 통해 약국 위치(위도, 경도) 조회")
@GetMapping("/findLoc")
@ResponseBody
public Map<String, Double> getLatitudeLongitude(@RequestParam("address") String address) {
    return pharmacyService.kakaoLocalAPI(address);
}
```

```
@Operation(summary="약국 정보 수정", description="약국 정보 수정")
@PostMapping("/updatePharmacyInfo")
public String updatePharmacyInfo(@ModelAttribute PharmacyDTO pharmacyDTO) {
    pharmacyService.updatePharmacyInfo(pharmacyDTO);
    return "redirect:/pharmacy";
}
```

Main work ⑩ 병원 / 약국 정보 삭제

- 관리자만 가능하며 병원 조회 페이지에서 병원 삭제 아이콘 클릭 시 병원 정보가 삭제됩니다
- 관리자만 가능하며 약국 조회 페이지에서 약국 삭제 아이콘 클릭 시 약국 정보가 삭제됩니다

```
@Operation(summary="병원 정보 삭제", description="병원 정보 삭제", hidden=true)
@PostMapping("/deleteHospital")
public ResponseEntity<?> deleteHospital(@RequestParam("hospitalId") long hospitalId) {
    hospitalService.deleteHospital(hospitalId);
    return ResponseEntity.ok().build();
}
```

```
@Operation(summary="약국 정보 삭제", description="약국 정보 삭제", hidden=true)
@PostMapping("/deletePharmacy")
public ResponseEntity<?> deletePharmacy(@RequestParam("pharmacyId") long pharmacyId) {
    pharmacyService.deletePharmacy(pharmacyId);
    return ResponseEntity.ok().build();
}
```

Main work ⑪ 병원 / 약국 북마크

- 병원 / 약국 조회 페이지에서 병원 / 약국 북마크 아이콘 클릭 시 북마크 탭에 병원 / 약국 정보가 추가됩니다
- 북마크 삭제 시 병원 / 약국 조회 페이지에서 다시 북마크 아이콘을 클릭하면 삭제됩니다

북마크 추가

병원 이름	주소	병원 유형	경신 날짜	북마크
중앙재활원의원	강원특별자치도 동해시 천곡로 52, (천곡동)	한의원	2024. 6. 26.	★
e플러스치과의원	강원특별자치도 강릉시 경강로 2120, 5층 (옥전동, 시내동)	치과의원	2024. 6. 26.	☆
본치과의원	강원특별자치도 강릉시 입암로 157, 2층 (입암동, 재일빌딩)	치과의원	2024. 6. 26.	★
유디치과의원	강원특별자치도 강릉시 금성로 60, 3,4층 (성내동, 보알빌딩)	치과의원	2024. 6. 26.	★



병원 / 사용자 프로필 / 북마크

병원	약국			
병원 이름	유형	주소	전화번호	홈페이지 주소
중앙재활의원	한의원	강원특별자치도 동해시 천곡로 52, (천곡동)	033-535-1138	http://www.jhaniwon.com
본치과의원	치과의원	강원특별자치도 강릉시 입암로 157, 2층 (입암동, 재일빌딩)	033-652-2228	https://
유디치과의원	치과의원	강원특별자치도 강릉시 금성로 60, 3,4층 (성내동, 보알빌딩)	033-655-6400	http://www.udh.co.kr

북마크 삭제

병원 이름	주소	병원 유형	경신 날짜	북마크
중앙재활원의원	강원특별자치도 동해시 천곡로 52, (천곡동)	한의원	2024. 6. 26.	★
e플러스치과의원	강원특별자치도 강릉시 경강로 2120, 5층 (옥전동, 시내동)	치과의원	2024. 6. 26.	☆
본치과의원	강원특별자치도 강릉시 입암로 157, 2층 (입암동, 재일빌딩)	치과의원	2024. 6. 26.	☆
유디치과의원	강원특별자치도 강릉시 금성로 60, 3,4층 (성내동, 보알빌딩)	치과의원	2024. 6. 26.	★



병원 약국

병원 이름	유형	주소	전화번호	홈페이지 주소
중앙재활원의원	한의원	강원특별자치도 동해시 천곡로 52, (천곡동)	033-535-1138	http://www.jhaniwon.com
유디치과의원	치과의원	강원특별자치도 강릉시 금성로 60, 3,4층 (성내동, 보알빌딩)	033-655-6400	http://www.udh.co.kr

Main work ⑪ 병원 / 약국 북마크

병원 북마크

```

@Service
public class HospitalBookmarkServiceImpl implements HospitalBookmarkService {

    @Autowired
    private HospitalBookmarkDAO hospitalBookmarkDAO;

    @Autowired
    private MemberDAO memberDAO;

    @Override
    public List<HospitalBookmarkDTO> getBookmarksForMember(String memberId) {
        if (memberId != null) {
            long memberId = memberDAO.getUserInfo(memberId).getMemberIdx();
            List<HospitalBookmarkDTO> bookmarkList = hospitalBookmarkDAO.findByMemberIdx(memberId);
            if (bookmarkList.size() > 0) {
                return bookmarkList;
            } else {
                return Collections.emptyList();
            }
        } else {
            return Collections.emptyList();
        }
    }

    @Override
    public void addHospitalBookmark(String memberId, long hospitalId) {
        long memberId = memberDAO.getUserInfo(memberId).getMemberIdx();
        List<HospitalBookmarkDTO> bookmarkList = hospitalBookmarkDAO.findByMemberIdx(memberId);
        boolean alreadyBookmarked = bookmarkList.stream().anyMatch(bookmark -> bookmark.getHospitalId() == hospitalId);
        if (!alreadyBookmarked) {
            HospitalBookmarkDTO newBookmark = new HospitalBookmarkDTO();
            newBookmark.setMemberIdx(memberId);
            newBookmark.setHospitalId(hospitalId);
            hospitalBookmarkDAO.addHospitalBookmark(newBookmark);
        }
    }

    @Override
    public void removeHospitalBookmark(String memberId, long hospitalId) {
        long memberId = memberDAO.getUserInfo(memberId).getMemberIdx();
        List<HospitalBookmarkDTO> bookmarkList = hospitalBookmarkDAO.findByMemberIdx(memberId);
        Optional<HospitalBookmarkDTO> bookmarkToRemove = bookmarkList.stream()
            .filter(bookmark -> bookmark.getHospitalId() == hospitalId).findFirst();
        bookmarkToRemove.ifPresent(bookmark -> {
            Map<String, Object> params = new HashMap<>();
            params.put("memberIdx", memberId);
            params.put("hospitalId", hospitalId);
            hospitalBookmarkDAO.removeHospitalBookmark(params);
        });
    }
}

```

약국 북마크

```

@Service
public class PharmacyBookmarkServiceImpl implements PharmacyBookmarkService {

    @Autowired
    private PharmacyBookmarkDAO pharmacyBookmarkDAO;

    @Autowired
    private MemberDAO memberDAO;

    @Override
    public List<PharmacyBookmarkDTO> getBookmarksForMember(String memberId) {
        if (memberId != null) {
            long memberId = memberDAO.getUserInfo(memberId).getMemberIdx();
            List<PharmacyBookmarkDTO> bookmarkList = pharmacyBookmarkDAO.findByMemberIdx(memberId);
            if (bookmarkList.size() > 0) {
                return bookmarkList;
            } else {
                return Collections.emptyList();
            }
        } else {
            return Collections.emptyList();
        }
    }

    @Override
    public void addPharmacyBookmark(String memberId, long pharmacyId) {
        long memberId = memberDAO.getUserInfo(memberId).getMemberIdx();
        List<PharmacyBookmarkDTO> bookmarkList = pharmacyBookmarkDAO.findByMemberIdx(memberId);
        boolean alreadyBookmarked = bookmarkList.stream().anyMatch(bookmark -> bookmark.getPharmacyId() == pharmacyId);
        if (!alreadyBookmarked) {
            PharmacyBookmarkDTO newBookmark = new PharmacyBookmarkDTO();
            newBookmark.setMemberIdx(memberId);
            newBookmark.setPharmacyId(pharmacyId);
            pharmacyBookmarkDAO.addPharmacyBookmark(newBookmark);
        }
    }

    @Override
    public void removePharmacyBookmark(String memberId, long pharmacyId) {
        long memberId = memberDAO.getUserInfo(memberId).getMemberIdx();
        List<PharmacyBookmarkDTO> bookmarkList = pharmacyBookmarkDAO.findByMemberIdx(memberId);
        Optional<PharmacyBookmarkDTO> bookmarkToRemove = bookmarkList.stream()
            .filter(bookmark -> bookmark.getPharmacyId() == pharmacyId).findFirst();
        bookmarkToRemove.ifPresent(bookmark -> {
            Map<String, Object> params = new HashMap<>();
            params.put("memberIdx", memberId);
            params.put("pharmacyId", pharmacyId);
            pharmacyBookmarkDAO.removePharmacyBookmark(params);
        });
    }
}

```

Main work ⑫ 커뮤니티

커뮤니티 생성

메인 / 커뮤니티



커뮤니티에 오신 것을 환영합니다.

커뮤니티에 참가하고 여러 사람들과 자유 주제로 토론해보세요.

커뮤니티 목록

공지사항

사이트 운영과 관련된 공지사항이 올라옵니다



8

Views

메인 / 커뮤니티 / 커뮤니티 추가

커뮤니티 이름*

인사게시판

작성 가능

커뮤니티 설명

커뮤니티를 찾아주셔서 감사합니다. 우리 모두 인사를 나눕니다

커뮤니티 목적*

회원들과의 첫인사

주소



- 커뮤니티 페이지에서 커뮤니티를 생성할 수 있습니다.
- 커뮤니티 생성 정보를 입력하고 추가 버튼을 클릭하면 커뮤니티 정보가 관리자에게 표시됩니다.
- 관리자가 커뮤니티 관리 페이지에서 커뮤니티를 승인하면 커뮤니티 목록에서 보여집니다.

커뮤니티명

설명

목적

상태

생성 일자

승인

공지사항

사이트 운영과 관련된 공지사항이 올라옵니다

공지사항 표시 커뮤니티

✓

2024-07-08

✓

인사게시판

커뮤니티를 찾아주셔서 감사합니다. 우리 모두 인사를 나눕니다

회원들과의 첫인사

✗

2024-07-10

승인

메인 / 커뮤니티



커뮤니티에 오신 것을 환영합니다.

커뮤니티에 참가하고 여러 사람들과 자유 주제로 토론해보세요.

커뮤니티 목록

공지사항

사이트 운영과 관련된 공지사항이 올라옵니다

인사게시판

커뮤니티를 찾아주셔서 감사합니다. 우리 모두 인사를 나눕니다



8

Views

0

Views

Main work ⑫ 커뮤니티

게시글 작성

메인 / 커뮤니티 / 인사게시판

커뮤니티를 찾아주셔서 감사합니다. 우리 모두 인사를 나눕시다

게시글 작성

Active Topics

Status

0

Members

0

Posts

게시글이 없습니다.

메인 / 커뮤니티 / 인사게시판 / 게시글 작성

게시글 제목*

안녕하세요

It's

게시글 내용

X B I S L | H | A | B | U | Ω | X | S |

B I S L | H | A | B | U | Ω | X | S |

안녕하세요. 잘 부탁드립니다.

body p

취소

작성

- 커뮤니티는 회원만 열람할 수 있습니다
- 커뮤니티 내에서 현재 참여하고 있는 회원 수, 게시글 수를 확인할 수 있습니다
- 게시글 작성 버튼을 통해 게시글을 작성할 수 있으며 게시글에 댓글을 추가할 수도 있습니다
- 게시글 수정 / 삭제는 작성자만 가능합니다

안녕하세요

2024년 07월 10일 14시 17분 It's 0

안녕하세요. 잘 부탁드립니다.

It's

댓글 내용을 입력해주세요

작성

Main work ⑫ 커뮤니티

게시글 수정 / 삭제

안녕하세요

2024년 07월 10일 14시 17분

안녕하세요. 잘 부탁드립니다.



안녕하세요

2024년 07월 10일 14시 17분

안녕하세요. 잘 부탁드립니다.^^

조

관리자

댓글 내용을 입력해주세요

작성

- 게시글 작성자라면 게시글 상세 화면에서 수정 아이콘을 클릭하여 게시글을 수정할 수 있습니다
- 게시글 작성자라면 게시글 상세 화면에서 삭제 아이콘을 클릭하여 게시글을 삭제할 수 있습니다.
- 게시글 내용 입력칸은 CKEditor를 사용하였습니다.

메인 / 커뮤니티 / 인사게시판 / 안녕하세요 / 게시글 수정

게시글 제목*

안녕하세요

it's

게시글 내용

CKEditor toolbar with various formatting options like bold, italic, underline, link, unlink, list, table, etc.

안녕하세요. 잘 부탁드립니다.^^

body p

원소

수정

Main work ⑫ 커뮤니티

커뮤니티 코드

```
@Service
public class CommunityServiceImpl implements CommunityService {

    @Autowired
    private CommunityDAO communityDAO;

    @Override
    public List<CommunityDTO> getCommunityList() {
        List<CommunityDTO> communityList = communityDAO.getCommunityList();
        return communityList.size() > 0 ? communityList : Collections.emptyList();
    }

    @Override
    public CommunityDTO findCommunity(long communityId) {
        return communityDAO.findCommunity(communityId);
    }

    @Override
    public CommunityDTO findCommunityByCommunityName(String communityName) {
        return communityDAO.findCommunityByCommunityName(communityName);
    }

    @Override
    public String validCommunityName(String communityName) {
        String getCommunityName = communityDAO.validCommunityName(communityName);
        return getCommunityName == null ? "y" : "n";
    }

    @Override
    public void createCommunity(CommunityDTO communityDTO) {
        communityDTO.setActiveYn("n");
        communityDTO.setTotalViewCount(0);
        communityDTO.setCreatedAt(new Date());
        communityDAO.createCommunity(communityDTO);
    }

    @Override
    public Map<String, Long> countActiveCommunity() {
        List<Map<String, Object>> results = communityDAO.countActiveCommunity();
        Map<String, Long> activeYnCountMap = new HashMap<>();

        for (Map<String, Object> result : results) {
            String activeYn = (String) result.get("ACTIVE_YN");
            Long count = ((Number) result.get("COUNT")).longValue();
            activeYnCountMap.put(activeYn, count);
        }
        return activeYnCountMap;
    }
}
```

게시글 코드

```
@Service
public class PostServiceImpl implements PostService {

    @Autowired
    private PostDAO postDAO;

    @Override
    public List<PostDTO> getPostList() {
        List<PostDTO> postList = postDAO.getPostList();
        return postList != null ? postList : Collections.emptyList();
    }

    @Override
    public List<PostDTO> getRecentPostList() {
        List<PostDTO> postList = postDAO.getRecentPostList();
        return postList != null ? postList : Collections.emptyList();
    }

    @Override
    public long getPostListCount(String communityName) {
        List<Map<String, Object>> postListCounts = postDAO.getPostListCount();
        for (Map<String, Object> postListCount : postListCounts) {
            if (communityName.equals(postListCount.get("COMMUNITY_NAME"))) {
                return Long.parseLong(postListCount.get("COUNT").toString());
            }
        }
        return 0;
    }
}
```

```
@Override
public void createPost(String communityName, PostDTO postDTO) {
    postDTO.setCommunityName(communityName);
    postDTO.setReplyCnt(0);
    postDTO.setViewCnt(0);
    postDTO.setCreatedAt(new Date());
    postDAO.createPost(postDTO);
}

@Override
public PostDTO findPostDetail(long postId) {
    PostDTO post = postDAO.findPostDetail(postId);
    postDAO.updateViewCnt(postId);
    return post;
}

@Override
public void updatePost(PostDTO postDTO) {
    postDTO.setUpdatedAt(new Date());
    postDAO.updatePost(postDTO);
}

@Override
public void deletePost(long postId) {
    postDAO.deletePost(postId);
}
```


Main work ⑫ 커뮤니티

댓글

안녕하세요

2024년 07월 10일 14시 17분 · 1인 · 0% · 0

안녕하세요, 잘 부탁드립니다. ^^

관리자

댓글 내용을 입력해주세요

작성

- 게시물 상세 화면에서 댓글을 작성할 수 있습니다.
- 게시물 상세 화면에서 본인이 작성한 댓글에 대해 수정 / 삭제가 가능합니다.

안녕하세요

2024년 07월 10일 14시 17분 · 1인 · 0% · 0

안녕하세요, 잘 부탁드립니다. ^^

Comments (1)



관리자

2024년 07월 10일 14시 55분

댓글 내용을 입력해주세요

삭제

수정



관리자

2024년 07월 10일 14시 55분

사이트를 방문해주셔서 감사합니다. 잘 부탁드립니다.

작성



관리자

2024년 07월 10일 15시 00분

사이트를 방문해주셔서 감사합니다. 잘 부탁드립니다.

삭제

수정

Main work ⑫ 커뮤니티

댓글

```
@Override
public void createReply(ReplyDTO replyDTO, String communityName) {
    replyDTO.setCommunityName(communityName);
    replyDTO.setCreatedAt(new Date());
    postDAO.updateReplyCntPlus(replyDTO.getPostId());
    replyDAO.createReply(replyDTO);
}
```

```
@Override
public void updateReply(ReplyDTO replyDTO) {
    replyDTO.setUpdatedAt(new Date());
    replyDAO.updateReply(replyDTO);
}
```

```
@Override
public void deleteReply(long postId, long replyId) {
    postDAO.updateReplyCntMinus(postId);
    replyDAO.deleteReply(replyId);
}
```

```
function deleteReply(replyId) {
    console.log(replyId);
    if (confirm("정말 이 댓글을 삭제하시겠습니까?")) {
        var communityName = $("#communityName").val();
        var postId = $("#postId").val();

        $.ajax({
            url: '/community/' + communityName + '/postDetail/' + postId + '/deleteReply',
            type: 'POST',
            contentType: 'application/x-www-form-urlencoded',
            data: { replyId: replyId },
            success: function(response) {
                alert("댓글이 삭제되었습니다.");
                location.reload();
            },
            error: function(xhr, status, error) {
                console.error('Error:', error);
                alert("댓글 삭제 중 오류가 발생했습니다.");
            }
        });
    }
}
```

```
function createReply() {
    var communityName = $("#communityName").val();
    var postId = $("#postId").val();
    var replyContent = $("#replyContent").val();
    var replyAuthor = $("#replyAuthor").val();

    let param = {
        "postId": postId,
        "nickname": replyAuthor,
        "content": replyContent
    }

    $.ajax({
        url: '/community/' + communityName + '/postDetail/' + postId + '/createReply',
        type: 'POST',
        contentType: 'application/json',
        data: JSON.stringify(param),
        success: function(response) {
            location.reload();
        }
    });
}
```

```
function showUpdateForm(replyId) {
    console.log(replyId);
    $("#showUpdateBtn" + replyId).hide();
    $("#replyContent" + replyId).hide();
    $("#deleteReplyBtn" + replyId).hide();
    $("#replyContentInput" + replyId).removeClass("d-none").addClass("mb-3");
    $("#replyUpdateBtn" + replyId).removeClass("d-none");
}
```

```
function updateReply(replyId) {
    var newContent = $("#replyContentInput" + replyId).val();
    if (newContent) {
        var communityName = $("#communityName").val();
        var postId = $("#postId").val();

        let param = {
            "postId": postId,
            "replyId": replyId,
            "content": newContent
        }

        $.ajax({
            url: '/community/' + communityName + '/postDetail/' + postId + '/updateReply',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify(param),
            success: function(response) {
                alert("댓글이 수정되었습니다.");
                location.reload();
            },
            error: function(xhr, status, error) {
                console.error('Error:', error);
                alert("댓글 수정 중 오류가 발생했습니다.");
            }
        });
    }
}
```

Main work 프로젝트 관리

- 일일 보고서 및 주간 보고서를 작성하여 현재 프로젝트 진행 상황을 검수자와 공유하였습니다.
- 이슈 리스트를 작성하여 프로젝트 진행 중 발생한 문제를 체계적으로 정리하고, 검수자로부터 조언을 받아 문제를 해결하였습니다.

일일 보고서

W 20240626_일일보고서_전우진.docx

W 20240627_일일보고서_전우진.docx

W 20240628_일일보고서_전우진.docx

W 20240629_일일보고서_전우진.docx

W 20240630_일일보고서_전우진.docx

W 20240701_일일보고서_전우진.docx

W 20240702_일일보고서_전우진.docx

W 20240703_일일보고서_전우진.docx

W 20240704_일일보고서_전우진.docx

W 20240705_일일보고서_전우진.docx

W 20240706_일일보고서_전우진.docx

W 20240707_일일보고서_전우진.docx

주간 보고서

W WR_01_주간보고서_전우진.docx

W WR_02_주간보고서_전우진.docx

W WR_03_주간보고서_전우진.docx

W WR_04_주간보고서_전우진.docx

이슈 리스트

프로젝트	amazon Web Service 기반 의료 서비스 제공 사이트 개발 프로젝트										작성자	전우진
시스템명	Mediquanta										작성일	2024-07-11
이슈 리스트 ID	구분	대분류	중분류	소분류	제수사항	접수일자	작성일자	처리현황	담당자	상태	완료일자	
ISSUE-D01	기능	USER	LAYOUT	HEADER	스크롤에 따라 헤더가 보이지 않는 이슈 발생	2024-06-11	전우진	헤더가 화면 상단에 고정되도록 css 수정	전우진	지정완료	2024-06-12	
ISSUE-D03	기능	USER	MEMBER	REGISTER	Postcode v2 사용 시 cross site warning 발생	2024-06-12	전우진	Postcode v2 js script 주소 변경	전우진	지정완료	2024-06-14	
ISSUE-T01	테스트	TEST	TEST	TEST	BCryptPasswordEncoder를 사용하여 패스워드를 암호화한 경우 equals로 비교할 시 false로 반환되어 비교 실패	2024-06-13	전우진	BCryptPasswordEncoder를 사용하여 패스워드를 암호화할 때마다 salt는 랜덤 다르게 생성되기 때문에 같은 입력 값에 대해 항상 같은 해시(7)가 생성되지 않으므로 equals 대신 match를 통해 비교	전우진	지정완료	2024-06-14	
ISSUE-P01	운영	PRODUCTION	AWS	ERROR	failed to start bean 'webserverstatop'는 재현되지 않음	2024-06-14	전우진	운영 환경에서 프로젝트 불러 실행 후 재실행	전우진	지정완료	2024-06-16	
ISSUE-P02	운영	PRODUCTION	AWS	ERROR	배드 시 Error resolving template는 재현되지 않음	2024-06-15	전우진	layout decorator "\${page}"으로 작성	전우진, 안병욱	지정완료	2024-06-16	
ISSUE-P03	운영	PRODUCTION	AWS	ERROR	redirect 속성에서 ?를 누락하여 실패도 실패 발생	2024-06-20	전우진	코드인 mapping 정보 수정 후 재배포	전우진	지정완료	2024-06-20	
ISSUE-D03	기능	USER	MEMBER	SENDTEMPERPASSWOR	임시 비밀번호 코드 전송 실패	2024-06-21	전우진	build grade refresh 후 작업	전우진	지정완료	2024-06-21	
ISSUE-D01	디자인	CSS	-	-	500px에서 css 적용 한상 발생	2024-06-24	전우진	css 경로 변경	전우진	지정완료	2024-06-24	
ISSUE-D04	기능	USER	HOSPITAL	SEARCH/HOSPITAL	병원 검색 시 검색 API가 지연됨	2024-06-26	전우진	검색 API의 throttling 설정 필요 여부 고려	전우진	보류	-	
ISSUE-D05	기능	USER	HOSPITAL	HOSPITALDETAIL	병원 상세 정보에서 지도 API 불연속 시 application.properties에 지정된 키 값을 가져오지 못하는 현상 발생	2024-06-26	전우진	application.properties에 지정하지 않고 직접 파일에 하드코딩 값을 가져오도록 변경	전우진	지정완료	2024-06-26	
ISSUE-D06	기능	USER	PROFILE	UPDATEPROFILE	회원 정보 편집 시 비밀번호로 검색 비밀번호를 등록 시 기존 비밀번호와 일치하는 비밀번호 확인	2024-06-29	전우진	회원정보에 대한 비밀번호는 암호화하지 않고 바로 등록 가능하도록 변경	전우진	지정완료	2024-06-29	
ISSUE-D07	기능	USER	BOOKMARK	HOSPITALBOOKMARK	책갈피 추가 후 페이지가 일치 않는 현상 발생	2024-07-03	전우진	책갈피 관련 javascript 코드 수정	전우진	지정완료	2024-07-03	
ISSUE-D08	기능	USER	COMMUNITY	CREATECOMMUNITY	커뮤니티 생성 시 제목이나 제목을 한글로만 www.kang.liga@kang.liga.com을 적용하여 해결	2024-07-05	전우진	커뮤니티 생성 주소 한글로만 URLEncoder.encode를 적용하여 해결	전우진	지정완료	2024-07-07	
ISSUE-P04	운영	PRODUCTION	AWS	ERROR	배드 시 Web server failed to start. Port 8080 was already in use라는 메시지가 출력	2024-07-07	전우진	일일방진 오류를 줄여줘서 재배포하여 해결	전우진	지정완료	2024-07-07	
ISSUE-M01	유지보수	SSL	-	-	AWS를 접속 시 403 에러로 보안 경고(HTTPS가 사용되지 않음)나타 내서 페이지 로딩도 무리 없을 부득이 조치함	2024-07-11	전우진	1. certbot을 통한 SSL 인증서 발급 후 재배포 시도 2. AWS 로드 밸런서 설정치 확인 (최근 추가된 부하량 보류)	전우진	보류	2024-07-11	

Main work 검수

단위 테스트와 통합 테스트를 통하여 검수를 진행하였습니다

단위 테스트

단위테스트 결과서

프로젝트명	Amazon Web Service 기반 의료 서비스 제공 사이트 개발 프로젝트	작성명	2024-05-31	테스트 수행명	2024-07-01
시스템명	Mediquest	작성자	전우진	승인자	인병욱

케이스	분류	체크항목	테스트 코드	결과
UT01	유계 > 회원가입	회원가입 정상동작 여부	<pre>@Order(1) @DisplayName("회원가입") @Test public void createMember() throws ParseException, IllegalStateException, IOException { MemberDTO memberDTO = new MemberDTO();</pre>	PASS

통합 테스트

순번	대분류	중분류	소분류	시나리오명	시나리오 개요	시나리오 흐름	케이스 번호	케이스	예상결과	결과
1		병원정보 등록	병원 정보 설정	병원 정보 등록	병원 정보를 등록할 수 있다	1. 관리자 계정으로 로그인 2. 헤더의 프로필 아이콘 클릭 3. 병원 관리 옆 불러오기 버튼 클릭	1	관리자 계정으로 로그인	로그인 성공 시 메인 화면으로 이동한다	PASS
						2	헤더의 프로필 아이콘 클릭	관리자 프로필 화면으로 이동한다	PASS	
						3	병원 관리 옆 불러오기 버튼 클릭	신규 병원 정보가 저장되며, 저장 완료 후 불러오기 버튼이 성공으로 표시된다	PASS	
2		병원정보 수정		병원 정보 수정	병원 정보를 수정할 수 있다	1. 관리자 계정으로 로그인 2. 헤더의 병원 조회 클릭 3. 병원 목록에서 수정 아이콘 클릭 4. 병원 정보 수정 페이지에서 수정 정보 입력 후 수정 버튼 클릭	1	관리자 계정으로 로그인	로그인 성공 시 메인 화면으로 이동한다	PASS
						2	헤더의 병원 조회 클릭	병원 조회 화면으로 이동한다	PASS	
						3	병원 목록에서 수정 아이콘 클릭	병원 수정 페이지로 이동한다	PASS	
						4	병원 정보 수정 페이지에서 수정 정보 입력 후 수정 버튼 클릭	병원 조회 화면에서 수정된 병원 정보를 확인할 수 있다	PASS	
3		병원정보 삭제		병원 정보 삭제	병원 정보를 삭제할 수 있다	1. 관리자 계정으로 로그인 2. 헤더의 병원 조회 클릭 3. 병원 목록에서 삭제 아이콘 클릭	1	관리자 계정으로 로그인	로그인 성공 시 메인 화면으로 이동한다	PASS
						2	헤더의 병원 조회 클릭	병원 조회 화면으로 이동한다	PASS	
						3	병원 목록에서 삭제 아이콘 클릭	병원 조회 화면에서 병원 정보가 삭제되었음을 확인할 수 있다	PASS	
4		약국정보 등록		약국 정보 등록	약국 정보를 등록할 수 있다	1. 관리자 계정으로 로그인 2. 헤더의 프로필 아이콘 클릭 3. 약국 관리 옆 불러오기 버튼 클릭	1	관리자 계정으로 로그인	로그인 성공 시 메인 화면으로 이동한다	PASS
						2	헤더의 프로필 아이콘 클릭	관리자 프로필 화면으로 이동한다	PASS	
				3	약국 관리 옆 불러오기 버튼 클릭	신규 약국 정보가 저장되며, 저장 완료 후 불러오기 버튼이 성공으로 표시된다	PASS			

제 포트폴리오를 검토해 주셔서 감사합니다.
