

맞춤형 복지 추천 서비스

Overview

새로운 서비스, 지원대상 및 조건들의 변경 등의 이유로
원하는 서비스를 찾기 어려워요



이에 따라 자신이 어떤 복지 혜택을 받을 수 있는지 잘 모르는 분들을 위해
나에게 맞는 복지 혜택을 추천해주는 서비스, 복지도를 제안합니다.

Service Strategy

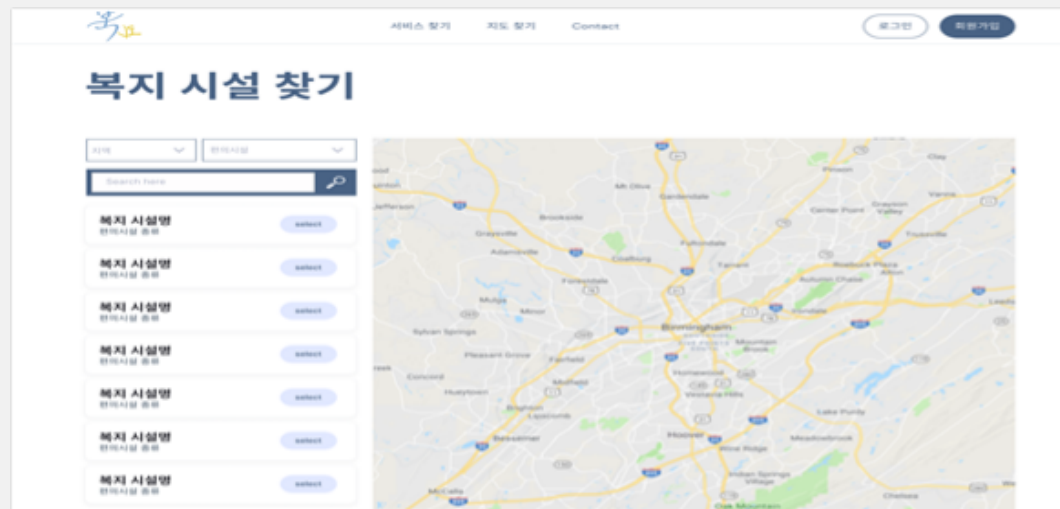
맞춤형 복지 추천 서비스, 복지도

나만의 맞춤형 복지 서비스



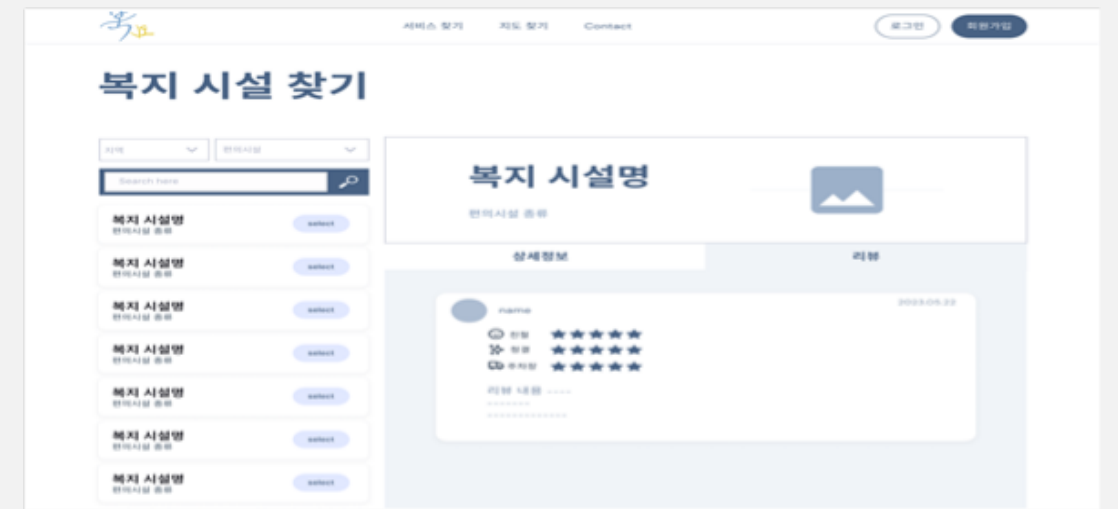
회원가입 시에 받은 정보를 활용해
개인에게 맞는 복지 서비스를
확인할 수 있습니다.

지도로 보는 복지건물



원하는 복지 시설의 위치를
지도로 손쉽게 파악하고
현재 운영여부 등의 정보를
쉽게 찾아볼 수 있습니다.

리뷰로 남기는 자세한 정보



복지 서비스에 대한 후기와
건물 리뷰 내용을 통해
친절도, 청결도, 주차 시설에 대한
자세한 정보를 확인할 수 있습니다.

Stack

Frontend



Backend



Cooperation



Frontend - Review, Map

Facility Review

😊 친절

★★★★☆

🌟 청결

★★★★☆

🚗 주차장

★★★★☆

지역

연령

이용한 서비스명

Search

리뷰를 작성해주세요.

완료

클릭하는 별의 위치에 따라 1~5점으로 저장

```
//클릭 이벤트 리스너 친절 등록
rateK.addEventListener('click',function(e){
    let elem = e.target;
    if(elem.classList.contains('rate_radio')){
        rating.setRate(elem.parentElement, parseInt(elem.value)); // setRate() 에 ".rating" 요소를 첫 번째 파라미터로 넘김
        storeK.push(elem.value);
    }
});
```

지도 페이지에서 마커 생성 및 지도에 표시



```
function addMarker(position, idx, title) {
    var imageSrc = 'https://t1.daumcdn.net/localimg/localimages/07/mapapidoc/marker_number_blue.png', // 마커 이미지 url, 스프라이트 이미지를 씁니다
        imageSize = new kakao.maps.Size(36, 37), // 마커 이미지의 크기
        imgOptions = {
            priteSize : new kakao.maps.Size(36, 691), // 스프라이트 이미지의 크기
            spriteOrigin : new kakao.maps.Point(0, (idx*46)+10), // 스프라이트 이미지 중 사용할 영역의 좌상단 좌표
            offset: new kakao.maps.Point(13, 37) // 마커 좌표에 일치시킬 이미지 내에서의 좌표
        },
        markerImage = new kakao.maps.MarkerImage(imageSrc, imageSize, imgOptions),
        marker = new kakao.maps.Marker({
            position: position, // 마커의 위치
            image: markerImage
        });
    marker.setMap(map); // 지도 위에 마커를 표출합니다
    markers.push(marker); // 배열에 생성된 마커를 추가합니다
    ...
    return marker;
}
```

Frontend - Address Management

도로명 주소를 입력 받았을 때 정해진 칸에 정보가 들어옴

```
//주소 메서드
function sample4_execDaumPostcode() {
    new daum.Postcode({
        oncomplete: function(data) {
            // 팝업에서 검색결과 항목을 클릭했을때 실행할 코드를 작성하는 부분.

            // 도로명 주소의 노출 규칙에 따라 주소를 표시한다.
            // 내려오는 변수가 값이 없는 경우엔 공백('')값을 가지므로, 이를 참고하여 분기 한다.
            var roadAddr = data.roadAddress; // 도로명 주소 변수
            var extraRoadAddr = ''; // 참고 항목 변수

            // 법정동명이 있을 경우 추가한다. (법정리는 제외)
            // 법정동의 경우 마지막 문자가 "동/로/가"로 끝난다.
            if(data.bname !== '' && /[동|로|가]$/g.test(data.bname)){
                extraRoadAddr += data.bname;
            }
            // 건물명이 있고, 공동주택일 경우 추가한다.
            if(data.buildingName !== '' && data.apartment === 'Y'){
                extraRoadAddr += (extraRoadAddr !== '' ? ', ' + data.buildingName : data.buildingName);
            }
            // 표시할 참고항목이 있을 경우, 괄호까지 추가한 최종 문자열을 만든다.
            if(extraRoadAddr !== ''){
                extraRoadAddr = ' (' + extraRoadAddr + ')';
            }

            // 우편번호와 주소 정보를 해당 필드에 넣는다.
            document.getElementById('sample4_postcode').value = data.zonecode;
            document.getElementById("sample4_roadAddress").value = roadAddr;
            document.getElementById("sample4_jibunAddress").value = data.jibunAddress;

            // 참고항목 문자열이 있을 경우 해당 필드에 넣는다.
            if(roadAddr !== ''){
                document.getElementById("sample4_extraAddress").value = extraRoadAddr;
            } else {
                document.getElementById("sample4_extraAddress").value = '';
            }

            var guideTextBox = document.getElementById("guide");
            // 사용자가 '선택 안함'을 클릭한 경우, 예상 주소라는 표시를 해준다.
            if(data.autoRoadAddress) {
                var expRoadAddr = data.autoRoadAddress + extraRoadAddr;
                guideTextBox.innerHTML = '(예상 도로명 주소 : ' + expRoadAddr + ')';
                guideTextBox.style.display = 'block';
            } else if(data.autoJibunAddress) {
                var expJibunAddr = data.autoJibunAddress;
                guideTextBox.innerHTML = '(예상 지번 주소 : ' + expJibunAddr + ')';
                guideTextBox.style.display = 'block';
            } else {
                guideTextBox.innerHTML = '';
                guideTextBox.style.display = 'none';
            }
        }
    }).open();
}
```

Daum Postcode Service - Chrome

about:blank

예) 판교역로 166, 분당 주공, 백현동 532

tip
아래와 같은 조합으로 검색을 하시면 더욱 정확한 결과가 검색됩니다.

도로명 + 건물번호
예) 판교역로 166, 제주 청단로 242

지역명(동/리) + 번지
예) 백현동 532, 제주 영광동 2181

지역명(동/리) + 건물명(아파트명)
예) 분당 주공, 연수동 주공3차

사서함명 + 번호
예) 분당우체국사서함 1~100

Powered by kakao | 우편번호

자택주소

우편번호 찾기

47342

부산 부산진구 가야대로668번

부산 부산진구 가야동 6-17

(가야동)

다솜재가복지시설

Backend - login

로그인이 정상적으로 되면 토큰을 발급하고 Refresh토큰은 쿠키에 넣어 보관

```
@Operation(summary = "로그인", description = "User의 로그인 기능 <br>로그인을 하고 토큰을 발급")
@PostMapping("/login")
public ResponseEntity<?> login(@RequestBody @Valid AuthDTO.LoginDto loginDto) {
    // User 등록 및 Refresh Token 저장
    AuthDTO.TokenDto tokenDto = authService.login(loginDto);

    // RT 저장
    HttpCookie httpCookie = ResponseCookie.from(name: "refresh-token", tokenDto.getRefreshToken())
        .maxAge(COOKIE_EXPIRATION)
        .httpOnly(true)
        .secure(true)
        .build();

    System.out.println("tokenDto.getAccessToken() = " + tokenDto.getAccessToken());
    System.out.println("tokenDto.getRefreshToken() = " + tokenDto.getRefreshToken());
    System.out.println("tokenDto.getGrantType() = " + tokenDto.getGrantType());
    System.out.println("tokenDto.getExpiresIn() = " + tokenDto.getExpiresIn());

    return ResponseEntity.ok()
        .header(HttpHeaders.SET_COOKIE, httpCookie)
        // AT 저장
        .header(HttpHeaders.AUTHORIZATION, tokenDto.getAccessToken())
        .build();
}
```

사용자가 요청할 때 마다 서버는 발급 된 토큰으로 사용자를 식별

```
public Claims getClaims(String token) {
    try {
        return Jwts.parserBuilder()
            .setSigningKey(signingKey)
            .build()
            .parseClaimsJws(token)
            .getBody();
    } catch (ExpiredJwtException e) { // Access Token
        return e.getClaims();
    }
}
```

토큰이 위조됐는지, 사용 기한이 만료되었는지 확인

```
public boolean validateRefreshToken(String refreshToken){
    try {
        if (redisService.getValues(refreshToken).equals("delete")) {
            return false;
        }

        Jwts.parserBuilder()
            .setSigningKey(signingKey)
            .build()
            .parseClaimsJws(refreshToken);

        return true;
    } catch (SignatureException e) {
        log.error("Invalid JWT signature.");
    } catch (MalformedJwtException e) {
        log.error("Invalid JWT token.");
    } catch (ExpiredJwtException e) {
        log.error("Expired JWT token.");
    } catch (UnsupportedJwtException e) {
        log.error("Unsupported JWT token.");
    } catch (IllegalArgumentException e) {
        log.error("JWT claims string is empty.");
    } catch (NullPointerException e){
        log.error("JWT Token is empty.");
    }

    return false;
}
```

Backend - error

필요한 응답 커스텀 가능하도록 Dto 설계

```
@Data
public class ErrorResponseDto {

    3 usages
    private int status;

    3 usages
    private String errorType;

    3 usages
    private String message;

    no usages  👤 SoMi Park
    public ErrorResponseDto(ErrorCode errorCode) {
        this.status = errorCode.getStatus();
        this.errorType = errorCode.getErrorType();
        this.message = errorCode.getMessage();
    }

    6 usages  👤 SoMi Park
    public ErrorResponseDto(ErrorCode errorCode, String message) {
        this.status = errorCode.getStatus();
        this.errorType = errorCode.getErrorType();
        this.message = message;
    }

    👤 SoMi Park
    @Override
    public String toString() {
```

ErrorCode를 커스텀하여 관리

```
@AllArgsConstructor
public enum ErrorCode{

    no usages
    AUTHENTICATION_FAILED( status: 401, errorType: "AUTHENTICATION-FAILED", m

    2 usages
    USER_NOT_FOUND_ERROR( status: 400, errorType: "USER-NOT-FOUND", message: "U

    2 usages
    BENEFITS_NOT_FOUND_ERROR( status: 400, errorType: "BENEFITS-NOT-FOUND", message: "BENEFITS NOT FOUND ERROR"),

    1 usage
    BENEFIT_REVIEW_NOT_FOUND_ERROR( status: 400, errorType: "BENEFIT-REVIEW-NOT-FOUND", message: "BENEFIT REVIEWS NOT FOUND ERROR"),

    no usages
    FACILITY_NOT_DELETABLE( status: 400, errorType: "FACILITY-NOT-DELETABLE", message: "FACILITY NOT DELETABLE"),

    1 usage
    INVALID_ENUM_VALUE( status: 400, errorType: "INVALID-ENUM-VALUE", message: "ENUM VALUE IS INVALID"),

    2 usages
    QNA_NOT_FOUND_ERROR( status: 400, errorType: "QNA-NOT-FOUND", message: "QNA NOT FOUND ERROR"),

    1 usage
```

예외를 중앙에서 관리하여 클라이언트에게 일관된 에러 응답을 제공

```
@RestControllerAdvice
public class GlobalExceptionHandler {

    no usages  👤 SoMi Park
    @ExceptionHandler(CustomException.class)
    public ResponseEntity<ErrorResponseDto> handleCustomException(CustomException exception) {
        ErrorResponseDto errorResponseDto = new ErrorResponseDto(exception.getErrorCode(), exception.getMessage());
        return new ResponseEntity<>(errorResponseDto, HttpStatus.valueOf(errorResponseDto.getStatus()));
    }

    no usages  👤 SoMi Park
    @ExceptionHandler(MethodArgumentTypeMismatchException.class)
    public ResponseEntity<ErrorResponseDto> handleMethodArgumentTypeMismatchException(MethodArgumentTypeMismatchException exception) {
        if(exception.getParameter().getParameterType().isEnum()){
            ErrorResponseDto errorResponseDto = new ErrorResponseDto(ErrorCode.INVALID_ENUM_VALUE, message: "Invalid Enum value");
            return new ResponseEntity<>(errorResponseDto, HttpStatus.valueOf(errorResponseDto.getStatus()));
        }

        ErrorResponseDto errorResponseDto = new ErrorResponseDto(ErrorCode.METHOD_ARGUMENT_TYPE_MISMATCHED, exception.getMessage());
        return new ResponseEntity<>(errorResponseDto, HttpStatus.valueOf(errorResponseDto.getStatus()));
    }

    no usages  👤 SoMi Park
    @ExceptionHandler(BindException.class)
    public ResponseEntity<ErrorResponseDto> handleBindException(BindException exception) {
        ErrorResponseDto errorResponseDto = new ErrorResponseDto(ErrorCode.PARAMETER_BINDING_ERROR, exception.getMessage());
    }
```


Cooperation

Swagger로 API 자동 관리



Issue 발행

2 Open 10 Closed

Author Label Projects Milestones Assignee Sort

- ☐ ☒ 복지 서비스 맞춤 추천 API 구현 API #21 opened 2 days ago by fsm12
- ☐ ☒ 복지 건물 리뷰 조회, 건물 탐색 API 구현 API #20 opened 3 days ago by Mintchodan

Project 관리

Done 8

This has been completed

- ☒ bokjido-backend #17
복지 서비스 조회, 맞춤 탐색, 리뷰 조회 API 구현
- ☒ bokjido-backend #9
시설 MVC 패턴 적용 및 생성, 조회 기능 구현
- ☒ bokjido-backend #18
시설 MVC 패턴 적용 및 생성, 조회 기능 구현
- ☒ bokjido-backend #19
Feat[#17] 복지 서비스 관련 API 구현
- ☒ bokjido-backend #22

2 Open 10 Closed

Author Label Projects Milestones Assignee Sort

- ☐ ☒ User Entity 항목 추가 API #23 by Mintchodan was closed 6 hours ago
- ☐ ☒ 질의응답 및 댓글 페이지 API 구현 API #22 by fsm12 was closed 7 hours ago
- ☐ ☒ 복지 서비스 조회, 맞춤 탐색, 리뷰 조회 API 구현 API #17 by fsm12 was closed 4 days ago

Pull Request

1 Open 6 Closed

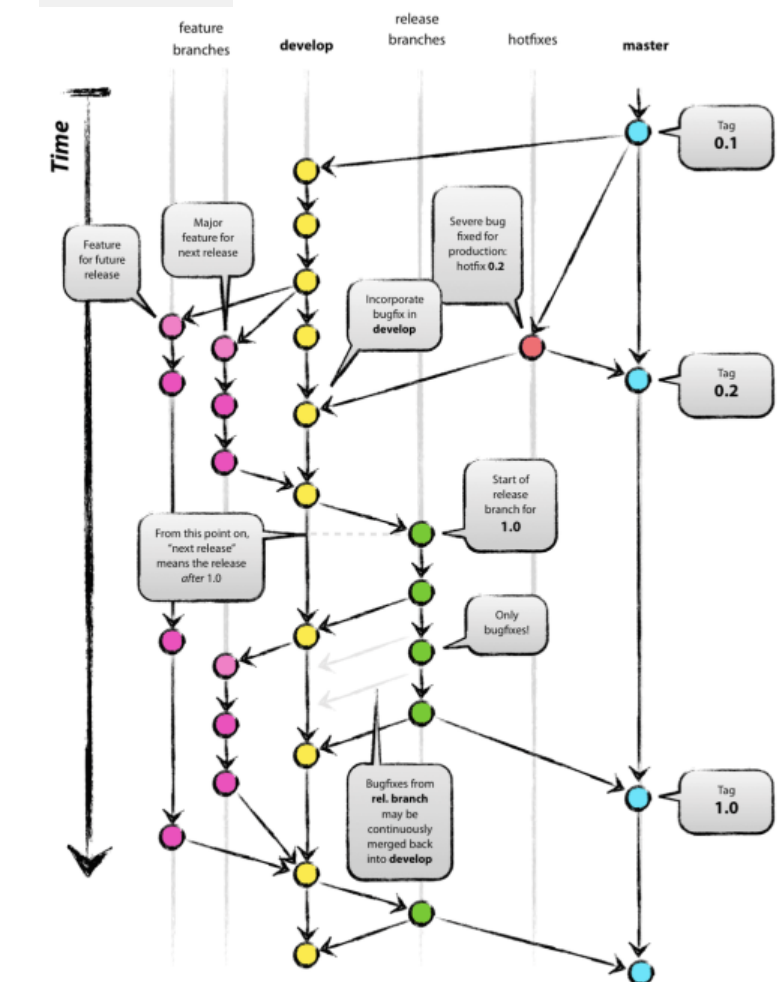
Author Label Projects Milestones Reviews Assignee Sort

- ☐ ☒ Feat[#22] 질의응답 게시판 및 댓글 API 구현 API #25 by fsm12 was merged 7 hours ago • Approved
- ☐ ☒ Feat[#23] UserEntity 항목 추가 API #24 by Mintchodan was merged 6 hours ago • Approved
- ☐ ☒ Feat[#17] 복지 서비스 관련 API 구현 API #19 by fsm12 was merged 4 days ago • Approved
- ☐ ☒ 시설 MVC 패턴 적용 및 생성, 조회 기능 구현 #18 by Mintchodan was merged last week • Approved
- ☐ ☒ 시설 MVC 패턴 적용 및 생성, 조회 기능 구현 #9 #16 by Mintchodan was closed last week • Approved
- ☐ ☒ Feature #1 시설 MVC 패턴 적용 및 생성, 조회 기능 구현 #8 by Mintchodan was closed last week

Commit Convention

feat: 새로운 기능 추가
fix: 버그 수정
docs: 문서
style: 디자인 변경
refactor: 코드 리팩토링
chore: 기타 수정

Git Flow



Demo Video

[서비스 찾기](#)[지도 찾기](#)[게시판](#)[로그인](#)[회원가입](#)

맞춤형 복지서비스는
회원가입을 통해 전달받은
정보들을 토대로 회원님께
맞는 정보들을 제공해드립니다.

맞춤형 복지서비스 편안한 삶을 누리보세요!



맞춤형 서비스

Team

박소미



Backend Developer

프로젝트 총괄

DB 설계

복지 서비스 관련 기능 구현

질의응답 및 댓글 기능 구현

신민승



Backend Developer

DB 설계

로그인, 회원 가입 기능 구현

JWT 활용
(발급, 갱신, 조회, 유효성 검사)

복지 시설 관련 기능 구현

임민욱



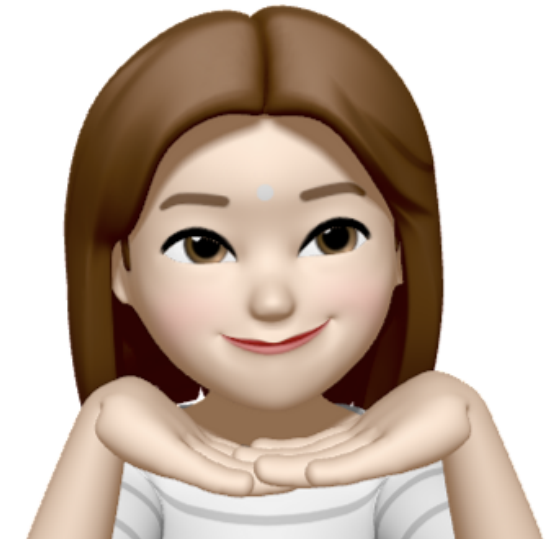
Frontend Developer

웹 사이트 메인 페이지

로그인 및 회원가입 페이지

마이페이지 구성

최윤성



Frontend Developer

웹사이트 디자인 구상

복지 서비스 및 시설의
조회, 상세조회, 리뷰 페이지 구현