

인터페이스 구현 보고서

소속 팀	B팀	작성자	소연희
------	----	-----	-----

인터페이스 명	기록 리스트 (프론트)	소요시간	2시간
구현 목표	사용자들이 작성한 기록들을 한 번에 확인할 수 있는 페이지로 사용자들의 기록을 공유하는 공간		
구현 내용	<p>총 기록 수 표시</p> <ul style="list-style-type: none">- 사용자들의 기록을 한 번에 확인할 수 있도록 상단에 총 도서기록 개수를 <code>totalResults</code> 값으로 화면에 출력 <p>정렬 기능</p> <ul style="list-style-type: none">- 기록의 정렬을 변경하기 위해 <code>date</code>와 <code>rating</code>으로 <code>sort</code>하여 최신순, 별점순으로 정렬 옵션 제공- 정렬 버튼 클릭 시, 상태를 갱신하고, <code>List.from()</code>으로 복사한 리스트를 정렬하여 출력 <p>기록 목록 조회 및 무한스크롤</p> <ul style="list-style-type: none">- 도서의 기록 목록을 조회하기 위해 <code>http.get</code>을 통해 서버에서 전체 기록 데이터 요청- <code>ScrollController</code>와 <code>scroll()</code>을 활용하여 페이지 하단 도달 시 다음 페이지 데이터 요청 후 리스트 추가 <p>기록 상세 확인 팝업</p> <ul style="list-style-type: none">- 각 콘텐츠 클릭 시 <code>ReviewDialog</code> 팝업을 호출하여 콘텐츠의 상세 내용들 확인 가능- 팝업 내 버튼을 클릭하면 디테일 페이지로 이동 <p>예외처리</p> <ul style="list-style-type: none">- 각 필드에 <code>null</code> 또는 빈값을 방지하기 위해 ?? 연산자를 통해 기본값 지정- 리스트가 없을 경우 <code>noList.png</code> 이미지를 표시 <p>사용한 위젯</p> <ul style="list-style-type: none">- <code>BookItem</code> : 각 기록 정보를 표시하는 위젯- <code>ReviewDialog</code> : 콘텐츠의 상세내용을 보여주기 위한 팝업- <code>ListView.builder</code> : 무한 스크롤 구현- <code>ScrollController</code> : 스크롤 이벤트 감지 및 페이지 증가 처리- <code>SvgPicture.asset</code> : 정렬 버튼에 SVG 아이콘 사용		
발생된 이슈 및 해결 방안	<ul style="list-style-type: none">- 기록 데이터가 여러 번 들어오는 이슈 발생 → <code>insert.php</code>에서 <code>book</code> 테이블에 도서 정보가 들어있는지 확인하는 구문에서 “<code>SELECT * FROM book WHERE book_id = '\$book_id'</code>” 에서 <code>id = '\$book_id'</code> 에서 <code>book_id = '\$book_id'</code>로 변경하여 처리하였습니다.		

인터페이스 명	기록 리스트 (서버)	소요시간	3시간
구현 목표	사용자들이 작성한 기록들을 한 번에 확인할 수 있는 페이지로 사용자들의 기록을 공유하는 공간 전체 기록 목록을 페이지 단위로 불러오는 PHP API		
구현 내용	<p>[사용 API]</p> <p><u>all_get.php</u></p> <ul style="list-style-type: none">- 사용자가 작성한 전체 도서 기록을 페이지 단위로 클라이언트에 전달- reviews테이블과 book 테이블을 LEFT JOIN 하여 책 정보와 함께 기록을 조회- 페이지당 10개의 기록을 최신순으로 조회- 전체 기록 수를 따로 구해 total로 반환하여 data안에 total값과 reviews 배열이 들어가 있는 형태로 응답받을 수 있음 <pre>{ "total ":20, "reviews ":[{ ...안에 reviews테이블과 book테이블 합친 데이터 }] }</pre> <ul style="list-style-type: none">- 프론트에서 해당 서버를 연동하기 위해 http.get(Uri.parse('\${bookapiUrl}/all_get.php?page=\$page'))로 서버에서 정보를 받아와 JSON 형태로 반환- 서버 연결 응답이 200일 경우 reviews 배열을 파싱하여 플러터 내부 리스트에 저장하고 1페이지 요청 시 total 값도 함께 저장, 200이 아닐 경우 콘솔에 에러 로그 출력		
발생된 이슈 및 해결 방안			

인터페이스 명	디테일 페이지 (프론트)	소요시간	4시간
구현 목표	도서의 상세 정보들과 해당 도서에 관련된 기록을 남길 수 있는 페이지로 사용자들이 독서 후 기록을 남기기 위함		
구현 내용	<p><u>도서 정보</u></p> <ul style="list-style-type: none">- 선택한 콘텐츠의 도서에 관한 정보들을 보여줌(제목, 줄거리, 출판사, 작가, 출판일에 대한 정보) <p><u>독서 기록</u></p> <ul style="list-style-type: none">- 사용자가 해당 도서에 남긴 기록을 확인할 수 있음- 기록의 총 수를 reviews.length 값을 통해 출력- 기록마다 수정과 삭제 버튼이 존재 <p>수정 : 수정 버튼 클릭 → 비밀번호 입력 팝업 → 비밀번호 일치 시, 기존 값이 입력된 입력 폼 열림 → 수정 후 저장 시, 서버에 요청하여 리스트 갱신</p> <p>삭제 : 삭제 버튼 클릭 → 비밀번호 입력 팝업 → 비밀번호 일치 시, 삭제 확인 팝업 열림 → 삭제 확인 시, 서버에 요청하여 해당 기록 삭제</p> <p><u>입력 폼</u></p> <ul style="list-style-type: none">- 상세페이지에서 도서에 관한 기록 작성 가능- 작성 시, 별점, 닉네임, 내용, 비밀번호를 입력해야 저장이 가능하도록 구현- 비밀번호를 기반으로 기록을 수정하거나 삭제가 가능하도록 함- 하단 흔적 남기기 버튼을 클릭 후, 입력 폼 작성 후 유효성 검사를 통해 저장- 이때, 동일한 닉네임인지 확인과 비밀번호가 영문, 숫자, 특수문자가 포함된 비밀번호인지 정규식 검사를 통과해야 저장이 가능 <p><u>사용 API</u></p> <ul style="list-style-type: none">- item_id.php : itemId 값을 기반으로 도서 정보를 조회하기 위함- review_get.php : 해당 도서에 작성된 모든 기록을 조회하기 위함- insert.php : 새로운 기록을 저장하기 위한 API- update.php : 기존 기록을 수정하기 위한 API- delete.php : 사용자가 작성한 기록을 삭제하기 위한 API- password_check.php : 기록 수정 삭제 시, 입력한 비밀번호가 해당 기록 작성자의 비밀번호와 일치하는지 확인하기 위함- check_nickname.php : 동일한 닉네임이 이미 존재하는지 확인하여 중복 방지를 하기 위함		
발생된 이슈 및 해결 방안			

인터페이스 명	디테일 페이지 (서버)	소요시간	8시간
구현 목표	도서의 상세 정보들과 해당 도서에 관련된 기록을 남길 수 있는 페이지로 사용자들이 독서 후 기록을 남기기 위함 해당 도서에 기록을 남기고, 수정 및 삭제를 할 수 있음		
구현 내용	<p>[사용 API]</p> <p><u>review_get.php</u></p> <ul style="list-style-type: none">- 요청방식 : Method-GET, 파라미터-bookId(필수:조회할 도서의 고유 ID)- 특정 도서에 작성된 모든 리뷰를 가져오는 api로 최신순으로 정렬되며, 닉네임, 내용, 별점, 작성일, id 등의 정보를 JSON 형식으로 반환- bookId를 기준으로 reviews 테이블에서 해당 도서에 대한 모든 리뷰 데이터를 조회- 조회된 결과를 반복문을 통해 reviews 배열로 구성하며, 관련 정보들을 포함하여 JSON 형식으로 반환 <pre>[{ "nickname": "booklover123", "content": "이 책 정말 감동적이에요.", "rating": 4.5, "date": "2024-12-01", "review_id": 12 }]</pre> <p><u>insert.php</u></p> <ul style="list-style-type: none">- 요청방식 : Method-POST, 필수 요청 필드-도서 정보(book_id, book_cover, book_title, book_author, book_publisher, book_pubDate), 기록 정보(rev_nickname, rev_password, rev_content, rev_rating)- 도서 상세페이지에서 사용자가 남긴 기록을 DB에 저장하고, 해당 도서가 book 테이블에 존재하지 않을 경우 자동으로 도서 정보를 등록- POST 요청에 도서 및 기록 정보를 수신- 비밀번호 유효성 검사 진행 : 영문, 숫자, 특수문자를 포함한 6자 이상의 비밀번호가 조건이며, 실패 시, 해당 부분이 포함되어야 한다는 메시지 반환- book 테이블에 해당 도서 id 가 존재하는지 확인하고, 존재하지 않으면 도서 정보를 DB에 새로 등록- reviews 테이블에 기록 정보를 저장- 저장 성공 여부에 따라 JSON 응답 반환(성공: { "success": true }, 실패: { "result": "error", "message": "에러메시지" }) <p><u>update.php</u></p> <ul style="list-style-type: none">- 요청방식 : Method-POST, 필수 요청 필드-review_id, password, rev_nickname, rev_content, rev_rating- 사용자가 작성한 기록을 수정할 수 있도록 하며, 비밀번호 검증을 통해 확인 절차 후 가능- POST 요청에서 데이터 수신 및 필수 항목 확인 시, review_id 또는 password 누락 시 요청 차단- DB에서 가져온 비밀번호와 비교 후 불일치 시, “비밀번호가 일치하지 않습니다” 라는 응답을 보냄- 입력된 닉네임, 내용, 별점, 날짜를 기준으로 해당 기록 정보 업데이트- 성공 시, { "success": true, "message": "리뷰가 성공적으로 수정되었습니다." } 반환- 실패 시, 오류 메시지 포함하여 { "success": false, "message": "수정 중 오류 발생: ..." } 반환 <p><u>delete.php</u></p> <ul style="list-style-type: none">- 요청방식 : Method-POST, 요청 필드-review_id, password- 사용자가 작성한 기록을 비밀번호 확인 후 삭제, 해당 도서에 기록이 한 개일 경우, 해당 도서 정보도 함께 삭제- 클라이언트로부터 JSON 형태의 데이터 수신- 필수값(review_id, password) 누락 시 오류 응답		

	<ul style="list-style-type: none">- review_id 기준으로 해당 기록의 비밀번호와 책아이디 조회하여 비밀번호가 일치할 경우, 해당 리뷰 삭제, 동일한 rev_book_id를 가진 다른 기록이 없는 경우, 해당 도서 정보도 삭제 <p>password_check.php</p> <ul style="list-style-type: none">- 요청방식 : Method-POST, 요청 필드-review_id, password- 기록 수정 또는 삭제 요청 전, 입력된 비밀번호가 해당 기록의 실제 비밀번호와 일치하는지 확인- JSON 형식의 요청 데이터 수신- 필수 입력값인 review_id와 password 유효성 검사 진행- review_id에 해당하는 리뷰 정보를 DB에서 조회하여 저장된 비밀번호와 입력된 비밀번호를 비교- 결과에 따라 메시지 반환 <p>check_nickname.php</p> <ul style="list-style-type: none">- 요청방식 : Method-POST, 요청 파라미터-rev_nickname- 기록 작성 시 사용자가 입력한 닉네임이 이미 사용중인지 확인- POST로 rev_nickname 값을 가져온 후, 빈 값일 경우 에러메시지를 반환- 닉네임이 존재하는지 DB 조회하여 개수가 1이상이면 중복, 0이면 사용 가능- 결과를 JSON 으로 반환 <pre>// 중복인 경우 { "result": "duplicate" } // 사용 가능한 경우 { "result": "available" } // 닉네임이 비어있는 경우 { "result": "error", "message": "닉네임이 비어있음" }</pre>
발생된 이슈 및 해결 방안	<ul style="list-style-type: none">- 기록 내용 수정 삭제 시, 비밀번호가 일치하는데 불일치하다는 메시지가 뜨는 상황, 비밀번호 입력 후 수정 시, 해당 기록 데이터에 값이 변경되는게 아니라 새로 추가되는 이슈 발생 → 서버에 보낼 데이터 중 칼럼 값이 누락되어 기능이 오작동하여, JSON body 구조를 재점검하고 필수 값을 모두 포함하도록 수정하였습니다.