d

|  |
| --- |
| **Project 3** |

|  |  |  |  |
| --- | --- | --- | --- |
| **2024년 2학기** | | | |
| 제출일 | | 2024. 12. 12. | |
| 과목명 | | 데이터베이스및데이터시각화 | |
| 담당교수 | | 이기훈 | |
|  | | | |
| 학과 | 컴퓨터정보공학부 | 학과 |  |
| 학번 | 2020202031 | 학번 |  |
| 이름 | 김재현 | 이름 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 학과 |  | 학과 |  |
| 학번 |  | 학번 |  |
| 이름 |  | 이름 |  |



1. 웹 애플리케이션에 대한 설명
2. 주요 기능 설명
3. **회원가입**

텍스트, 스크린샷, 손목시계이(가) 표시된 사진

자동 생성된 설명

메인페이지에서 회원가입 버튼을 누르면 회원가입 페이지로 이동합니다.

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

정보를 입력한 후 회원가입 버튼을 누르면 회원가입이 완료됩니다.

이미 존재하는 아이디로 가입하려고 하면, 다음과 같이 오류메시지를 출력합니다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

입력하지 않은 칸이 존재한다면 다음과 같이 경고메시지가 출력됩니다.

또한 이메일 주소로 도메인을 입력하지 않았을 때 역시 경고메시지가 출력됩니다.

1. **마이페이지(유저)**

텍스트, 스크린샷, 음식, 카탈로그이(가) 표시된 사진

자동 생성된 설명

메인페이지에서 마이페이지를 클릭하면 마이페이지로 이동합니다.

**[나의 정보 보기]**

텍스트, 스크린샷, 디스플레이, 번호이(가) 표시된 사진

자동 생성된 설명

왼쪽 네비게이션 바에는 정보 수정, 판매 내역, 구매 내역, 찜한 상품이 있고, 이를 클릭하면 해당 정보를 볼 수 있는 페이지로 이동할 수 있습니다.

중앙에는 회원가입할 때 입력한 사용자 정보가, 오른쪽에는 프로필 사진이 보여집니다.

회원가입할 때, 닉네임과 프로필 사진은 입력하지 않았으므로, 나의 정보 수정에서 수정 및 변경할 수 있습니다.

**[나의 정보 수정]**

텍스트, 스크린샷, 디스플레이, 번호이(가) 표시된 사진

자동 생성된 설명

나의 정보 수정 페이지로 이동하여, 닉네임과 프로필 사진을 입력한 후, 프로필 저장을 클릭하면 아래 이미지와 같이 닉네임과 프로필 사진이 업데이트 되는 것을 확인할 수 있습니다.

텍스트, 스크린샷, 포유류, 고양이이(가) 표시된 사진

자동 생성된 설명

닉네임과 프로필 사진이 잘 업데이트 된 모습입니다.

**[나의 판매 내역]**

텍스트, 스크린샷, 소프트웨어, 번호이(가) 표시된 사진

자동 생성된 설명

나의 판매 내역 페이지로 이동하면 유저가 판매하는 상품 리스트가 나옵니다.

상단에는 아직 판매 중인 상품이, 하단에는 이미 판매 완료된 상품이 보입니다.

비교적 최근에 업로드된 게시글이 상단에 위치하는 것을 확인할 수 있습니다.

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

또한 한 페이지에 상품이 5개씩 보이게 되며, 5개를 초과할 시, 나머지 상품은 다음 페이지로 이동하여 볼 수 있습니다. 다음 페이지 버튼을 누르면 다음 페이지로 넘어가며, 이전 페이지를 누르면 다시 이전 페이지로 돌아갑니다.

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

상품 리스트 중 하나를 클릭하면 다음과 같이 상품판매 게시글로 이동해서 세부내용을 확인 및 수정, 삭제할 수 있습니다.

**[나의 구매 내역]**

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

나의 구매 내역 페이지에서는 내가 구매한 상품들의 리스트를 볼 수 있습니다. 나의 판매 내역과 마찬가지로 최신순으로 정렬돼 있으며, 상품을 클릭하면 해당 판매게시글로 이동하게됩니다. 한페이지에 5개의 상품만이 보여지며 다음 페이지로 이동하여 나머지 상품을 확인할 수 있습니다.

**[내가 찜한 상품]**

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

내가 찜한 상품 페이지도 나의 판매 내역과 동일하게 판매 중인 상품과 판매 완료된 상품으로 나뉘어 있으며, 최신순으로 정렬돼있습니다. 한페이지에 5개의 상품만이 보여지며 다음 페이지로 이동하여 나머지 상품을 확인할 수 있습니다. 상품을 클릭하면 해당 판매게시글로 이동합니다.

1. **마이페이지(관리자)**

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

관리자 계정으로 로그인 후, 메인페이지에서 마이페이지를 클릭하면 다음과 같이 신고 당한 사용자 리스트를 보여줍니다.

신고 횟수 기준으로 내림차순 정렬합니다.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

신고 당한 사용자 중 한 명을 클릭하면, 다음과 같이 해당 유저의 정보와 신고 당한 판매 글이 보여집니다.

회원 탈퇴 버튼을 클릭하면 해당 유저가 usr테이블에서 삭제되면서 탈퇴처리됩니다.

판매 글 중 하나를 클릭하면 해당 판매글 페이지로 이동합니다.

x 버튼을 클릭하면 다시 신고 당한 유저 리스트가 보여집니다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

다음과 같이 삭제 성공이라는 메시지가 출력됩니다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

확인 버튼을 누르면 다음과 같이 세 명이었던 유저가 두 명으로 줄어든 것을 확인할 수 있습니다.

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

판매 글 중 하나를 클릭하자 판매글로 이동한 상황입니다.

관리자는 해당 판매글을 작성하지 않았지만 관리자 권한을 가지고 있기 때문에, 해당 판매글이 문제가 된다고 판단하면, 삭제 조치할 수 있습니다.

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

확인을 누릅니다.

텍스트, 고양이, 스크린샷이(가) 표시된 사진

자동 생성된 설명

자전거 관련 판매 글이 삭제되어 사라진 것을 확인할 수 있습니다.

1. 특별 기능 설명
2. SQL 설명
3. **회원가입**

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

INSERT INTO usr (Uname, Age, Sex, Phone, Email, Address, Id, Passwd)

VALUES(?, ?, ?, ?, ?, ?, ?, ?);

회원가입 페이지에서 유저는 이름, 나이, 성별, 전화번호, 이메일, 주소, 아이디, 비밀번호 정보를 입력하게 됩니다. 이 정보들을 서버로 전송하여 INSERT명령어를 통해 usr테이블에 저장함으로써 회원가입을 구현했습니다.

1. **마이페이지(유저)**
2. **[나의 정보 보기]**

SELECT \*

FROM usr

WHERE Id = ?;

usr테이블에서 PK는 Id이므로, Id를 통해 유저를 구분합니다. 따라서 현재 로그인 중인 유저의 Id를 통해 해당 유저의 정보만 SELECT합니다.

1. **[나의 정보 수정]**

프로필 사진 업로드 시:

UPDATE usr

SET Uname = ?, Age = ?, Sex = ?, Phone = ?, Email = ?, Address = ?, Passwd = ?,

Nickname = ?, Profile\_image = ?

WHERE id = ?;

프로필 사진 미업로드 시:

UPDATE usr

SET Uname = ?, Age = ?, Sex = ?, Phone = ?, Email = ?, Address = ?, Passwd = ?,

Nickname = ?

WHERE id = ?;

아이디를 제외한 모든 정보는 수정될 수 있습니다. 따라서 사용자가 수정한 데이터들을 통해 usr 테이블을 UPDATE합니다.

이 때, 프로필 사진 업로드 유무에 따라 다른 sql문을 사용한 이유는, 프로필 사진 미업로드 시 프로필 사진을 null로 업데이트하면, 기존에 존재하던 프로필 사진이 소멸되는 문제가 발생했기 떄문입니다.

1. **[나의 판매 내역]**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

SELECT \* FROM item\_datas WHERE Uid = ? ORDER BY Reg\_date DESC;

item\_datas 테이블에서 상품 게시자는 Uid입니다. 내가 판매한 상품을 구분하기 위해선 Uid가 나의 Id이면 됩니다.

따라서 WHERE 조건문을 통해 item\_datas 테이블에서 Uid가 나의 아이디인 것을 뽑아냈습니다. ORDER BY를 통해 게시일 내림차순으로 정렬 후 프론트앤드로 전송하여 출력했습니다.

1. **[나의 구매 내역]**

SELECT \* FROM item\_datas WHERE B\_uid = ? ORDER BY Reg\_date DESC;

item\_datas 테이블에서 상품 구매자는 B\_uid입니다. 내가 구매한 상품을 구분하기 위해선 B\_uid가 나의 Id이면 됩니다.

따라서 나의 판매 내역과 동일하게 sql문을 작성하되, WHERE 조건문만 B\_uid로 바꿔서 사용했습니다.

1. **[내가 찜한 상품]**

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

SELECT \*

FROM item\_datas

WHERE Ino IN (

SELECT Ino

FROM user\_like\_items

WHERE Uid = ?)

ORDER BY Reg\_date DESC;

user\_like\_items는 유저와 상품 간 좋아요 관계를 나타내는 관계테이블입니다. 어떤 유저가 어떤 상품에 좋아요를 남겼는지 판단할 수 있습니다.

내부 쿼리문은 WHERE Uid=? 통해 내가 좋아요를 남긴 상품들의 Ino을 SELECT합니다.

외부 쿼리문은 IN을 통해 item\_datas테이블에서 내가 좋아요를 남긴 상품들만 SELECT해줍니다.

ORDER BY를 통해 게시일 내림차순으로 프론트앤드로 보내 출력합니다.

1. **마이페이지(관리자)**
2. **[신고 당한 유저 리스트]**

SELECT Id, Rp\_cnt, Avg\_rating

FROM usr

WHERE Rp\_cnt > 0

ORDER BY Rp\_cnt DESC;

usr테이블에서 Rp\_cnt는 해당 유저가 신고 당한 횟수입니다.

따라서 Rp\_cnt > 0 즉, 신고 당한 유저를 골라내고, 신고 당한 횟수 순으로 내림차순 정렬하여 프론트앤드로 보내 출력합니다.

1. **[신고 당한 유저 정보]**

SELECT \*

FROM usr

WHERE Id = ?;

나의 정보 보기와 완전히 동일합니다.

1. **[신고 당한 유저의 판매 글]**

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

SELECT Ino, Title, Reg\_date, Report\_cnt

FROM item\_datas NATURAL JOIN (

SELECT Ino, COUNT(\*) AS Report\_cnt

FROM report JOIN item\_datas USING (Ino)

GROUP BY Ino) AS rp

WHERE Uid = ?

ORDER BY Report\_cnt DESC

LIMIT 5;

report는 상품과 상품을 신고한 유저 간 관계 테이블입니다.

내부 쿼리는 report와 item\_datas를 Ino로 JOIN하여 신고가 접수된 상품들의 정보를 뽑아내고, Ino로 GROUP BY를 함으로써 상품들의 Ino와 신고 당한 횟수를 뽑아냅니다.

외부 쿼리는 이렇게 뽑아낸 (Ino, Report\_cnt)를 다시 item\_datas와 Ino로 JOIN하여 신고 당한 판매글의 정보 + Report\_cnt까지 알아냈습니다.

LIMIT를 통해 최신 신고 글 5개만 보여줍니다.

1. **[회원 탈퇴시키기]**

DELETE

FROM usr

WHERE Id = ?;

유저의 Id를 통해 usr테이블에서 해당 유저를 DELETE함으로써 회원탈퇴 기능을 수행했습니다.

1. 데이터 시각화
2. YouTube 및 GitHub 링크

* 성적 마감시까지 공개로 설정

1. 고찰

저는 이번 팀프로젝트에서 아래와 같은 페이지들을 담당했습니다.

1. 회원가입
2. 마이페이지(유저)
3. 나의 정보 보기
4. 나의 정보 수정
5. 나의 판매 내역
6. 나의 구매 내역
7. 내가 찜한 상품
8. 마이페이지(관리자)
9. 신고 당한 유저 리스트 보기
10. 신고 당한 유저 탈퇴시키기
11. 신고 당한 유저 글 목록 보기

웹페이지의 기능 대부분이 회원가입을 통해 로그인을 해야만 사용할 수 있는 것들이라, usr테이블에 유저 정보를 입력하는 것이 필수적이었습니다. 이에, 회원가입 페이지를 빠르게 구현하여 팀원들이 쿼리 명령어를 통해 DB에 직접 데이터를 삽입하는 것이 아닌, 회원가입 페이지 UI를 통해 회원정보를 삽입할 수 있도록 하여, 개발 및 테스트를 간편하고 빠르게 할 수 있도록 도왔습니다.

그 이후에도 마감기한 전까지 시간을 적절히 분배하여 마이페이지의 다양한 기능들을 구현했습니다.

사실 INSERT, DELETE, UPDATE와 같은 쿼리문을 배우고 있는 시점에서 스키마를 작성했기에, 올바르게 잘 작성하고 있는 것인지 의문이 들기도 했으나, 이후에 ER to Relational 과 Database Design Theory에 대해 배우게 되면서, 생각보다 잘 작성했다는 느낌을 받았습니다.

그럼에도 item\_datas 테이블의 B\_uid column은 찝찝한 부분이 있었습니다. 유저와 상품 간 구매 관계는 N:1 relationship으로, usr의 PK인 Id를 item\_datas의 FK로 하는 것이 타당했으나, NULL값을 가지는 B\_uid가 많아서, 제 3의 테이블을 생성하여 구매 관계를 나타내는 것이 어땠을까 라는 고찰을 해봤습니다.