



포트폴리오



이지원
(1995.09.01)

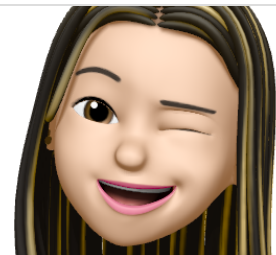
E-mail
깃허브

95jiwon91@naver.co

EZ195 - Overview

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to

<https://github.com/EZ195>



내 프로젝트 리스트

MEMO

자신만의 메모를 사진과 함께 기록하고 리스트를 확인해 볼 수 있는 메모장 프로젝트

<https://github.com/EZ195/Memo>

EZONISTA

게시글 업로드, 좋아요, 댓글 달기 기능을 갖춘 SNS 프로젝트

<https://github.com/EZ195/Ezonista>



취미를 통해 사람을 연결하는 커뮤니티

<https://github.com/EZ195/TheLink>

나와 똑같은 취미를 공유하는 사람을 주변에서 찾는 것은 쉬운 일이 아니다. 더욱이 설령 관심이 있더라도 혼자서 시작하기 어려울 수도 있다. 더링크는 취미를 바탕으로 사용자가 모임을 직접 주최하기도 하고 다른 사람의 모임에 참여하면서 같은 취미를 공유할 수 있도록 도와주는 커뮤니티이다. 관심 있는 분야를 처음 도전해보고 싶지만 막상 혼자서는 시작해볼 용기가 나지 않는 사람들, 혹은 혼자만 즐기던 취미를 다른 사람들과 공유하고 싶은 사람들의 니즈를 채워주기 위한 프로젝트이다.

프로젝트를 진행하며...

해당 프로젝트를 진행하며 개발 시작 전 설계 단계의 중요성에 대해 많이 느꼈다.

프로젝트 초반에 나름 꼼꼼하게 DB를 설계했다고 생각했다. 그러나 프로젝트 진행 도중 닉네임 수정 기능을 만들며 'userBO'와 'profileBO'가 순환참조 오류가 발생을 했다. 해당 오류를 해결하기 위해 user 테이블에 있던 nickname을 profile 테이블로 옮겼던 경험이 있다. 다행히 개발 초반 단계라 테이블 수정이 코드에 큰 영향을 미치지 않았지만 만약 개발 후반부에서 유사한 일이 발생했다면 더 많은 코드를 대대적으로 수정해야 했을 것이다.

하지만 한편으로는 설계에 대한 유연적 태도도 필요하다는 것을 느꼈다.

미리 프로젝트를 설계하면 해당 설계를 따라 그대로 기능들을 만들기만 하면 된다고 생각했다. 그래서 프로젝트를 금방 완성할 수 있을 것이라고 생각했다. 하지만 개발 중간중간 만들다 보니 새로운 기능을 추가하기도 하고 기능의 의도를 약간 바꾸는 등 생각보다 많은 수정들이 발생했다.

개발 능력 측면에서도 많은 향상이 있었다.

기능 구현의 응용력이 프로젝트를 시작하기 전보다 많이 향상되었다고 느꼈다. THE LINK 프로젝트를 시작하기 전에는 한 기능을 구현하면 그 기능만 구현할 줄 알았는데 프로젝트를 진행하다보니 해당 기능보다 조금 응용된 기능도 혼자 고민해보면서 스스로 구현할 수 있게되었다.

예상치 못한 오류가 정말 많다고 생각했다. 회원가입 관련 작업은 'EZONISTA' 프로젝트를 하면서도 만들어본 기능이었고, 다른 곳에서도 많이 해본 기능이기에 프로젝트 타임라인을 계획할 때 하루면 완성할 수 있을거라고 예상했다, 하지만 생각보다 많은 오류들이 발생하며 예상시간보다 많은 시간이 걸려서 회원가입 기능을 완성할 수 있었다. 프로젝트를 진행하며 기능 구현에 가장 많은 시간을 투자한 것은 회원가입 기능인 것 같다. 구현하는 기능마다 얼마나 걸릴지는 알기 힘들다고 느꼈다.

또한, 코드의 오류를 파악하는 능력도 향상되었다. 프로젝트를 진행하며 무수히 많은 오류들을 직면하며 오류의 종류들을 파악하고 이를 깃허브의 이슈사항에도 기재하며 나만의 오류 데이터베이스를 만들었다. 그에 이제는 어느정도 자주 나오는 오류들은 바로 대략 코드의 어떤 부분에서 오류가 났는지 가늠할 수 있게 되었다.

또한, css 및 부트스트랩에 대한 이해도가 증가되었다. 내가 직접 프로젝트의 모든 부분을 만들다 보니 프론트엔드의 영역 또한 내가 직접 해야했다. 이에 필요한 css 및 부트스트랩 기능을 검색을 통해 혼자 공부해보고 적용시켰다. 이 과정을 통해 다양한 css와 부트스트랩에 대해 어느 정도 수준의 이해도를 갖추게 되었다.

기술정보

tomcat, Mysql, spring framework, Spring boot, Mybatis, JSTL, Bootstrap, jQuery, java, javascript, html

기획

<https://ovenapp.io/project/A7f353wQ1G9M1HElunaXgJ7Xgn3l48DG#jBjrN>

프로젝트 설계

1. db 설계

<https://docs.google.com/spreadsheets/d/1tvTfUghExmRRobg8bVFAnLEsTj6UBUfFjxusoX6ZcQw/edit#gid=0>

테이블 : user				
설명 : 사용자 정보 테이블				
컬럼명	타입	NULL 가능 여부	AUTO_INCREMENT	설명
id	int	N	Y	primary key
loginEmail	varchar(64)	N	N	사용자 로그인 Email, 중복 체크 필요
password	varchar(64)	N	N	비밀번호, 암호화한 결과
bitrhYear	int	N	N	사용자 생일 연도
birthMonth	int	N	N	사용자 생일 월
birthDay	int	N	N	사용자 생일 일
gender	varchar(8)	N	N	사용자 성별
createdAt	timestamp	N	N	가입 날짜
updatedAt	timestamp	N	N	수정 날짜
테이블 : user_interest				
설명 : 사용자 지정 관심사 테이블				
컬럼명	타입	NULL 가능 여부	AUTO_INCREMENT	설명
id	int	N	Y	primary key
userId	int	N	N	사용자 ID (pk)
userInterest	varchar(16)	N	N	사용자 취미
createdAt	timestamp	N	N	생성 날짜

테이블 : profile				
설명 : 사용자 프로필 정보 테이블				
컬럼명	타입	NULL 가능 여부	AUTO_INCREMENT	설명
id	int	N	Y	primary key
userId	int	N	N	프로필 사용자 ID (pk)
nickname	varchar(32)	N	N	프로필 사용자 닉네임, 중복확인 필요
introduce	text	Y	N	프로필 사용자 자기소개
profileImagePath	varchar(256)	Y	N	프로필 사용자 이미지
createdAt	timestamp	N	N	생성 날짜
updatedAt	timestamp	N	N	수정 날짜
테이블 : follow				
설명 : 사용자 별 팔로우 및 팔로워 테이블				
컬럼명	타입	NULL 가능 여부	AUTO_INCREMENT	설명
id	int	N	Y	primary key
followerId	int	N	N	팔로우 하는 사용자의 ID (pk)
followerNickname	varchar(32)	N	N	팔로우 하는 사용자의 닉네임
followeeId	int	N	N	팔로우 받는 사용자의 ID
followeeNickname	varchar(32)	N	N	팔로우 받는 사용자의 닉네임
createdAt	timestamp	N	N	생성 날짜

2. url 설계

<https://docs.google.com/spreadsheets/d/1tvTfUghExmRRobg8bVFAnLEsTj6UBUfFjxusoX6ZcQw/edit#gid=1115453003>

View URL			
제목	URL	parameter	설명
로그인	/user/singin_view		로그인 화면
회원가입	/user/singup_view		회원가입 화면
프로필	/user/profile/profile_view	userId : 사용자 ID	사용자 프로필 화면
프로필 수정	/user/profile/update_view	userId : 사용자 ID	사용자 프로필 수정 화면
내 모임 리스트	/user/myjoin_list_view	userId : 사용자 ID	사용자가 참여한 모임 리스트 화면
주최 모임 리스트	/user/myhost_list_view	userId : 사용자 ID	사용자가 주최한 모임 리스트 화면
참여 신청자 리스트	/post/join/approve_list_view	userId : 작성자 ID postId : 게시물 ID	모임에 참여 신청을 한 사용자 리스트 화면 게시글 작성자만 볼 수 있음
게시글 작성	/post/create_view		게시글 작성화면
게시글 수정	/post/update_view	postId : 게시물 ID	작성된 게시물 수정 화면
게시글 상세보기	/post/detail_view	postId : 게시물 ID	게시글 상세정보 화면
타임라인	/post/timeline_view		게시글 및 사용자 리스트 화면
검색 페이지	/search/search_view		사용자 및 게시물 키워드로 검색한 결과를 보여주는 화면
모임 그룹채팅 리스트	/chat/list_view		사용자가 참가하는 모임의 단톡방 리스트 화면
채팅창	/chat/detail_view		단톡방 채팅창 화면

3. 일정설계

단계		세부정보	완료 여부	4월																														5월				
				5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5				
1	구상	프로젝트 기획	☑																																			
2	설계	DB 설계	☑																																			
	설계	URL 설계	☑																																			
3	구현	회원가입 기능 구현	☑																																			
	구현	관심사 저장 기능 구현	☑																																			
	구현	로그인 기능 구현	☑																																			
	구현	로그아웃 기능 구현	☑																																			
	구현	프로필 생성 기능	☑																																			
	구현	프로필 수정 기능 구현	☑																																			
	구현	게시물 글쓰기 기능 구현	☑																																			
	구현	게시물 리스트 구현	☑																																			
	구현	게시물 세부화면 구성	☑																																			
	구현	게시물 수정 기능 구현	☑																																			
	구현	게시물 삭제 기능 구현	☑																																			
	구현	모임 신청/취소 기능	☑																																			
	구현	참여자 승인 기능	☑																																			
	구현	후기 작성 기능 구현	☑																																			
	구현	후기 수정 기능 구현	☑																																			
	구현	후기 삭제 기능 구현	☑																																			
	구현	팔로우/언팔로우 기능 구현	☑																																			
	구현	해시태그 검색 기능 구현	☑																																			
	구현	키워드 검색 기능 구현	☑																																			
	구현	캘린더 기능 구현	☑																																			
4	테스트	회원가입 및 로그인 테스트	☑																																			
	테스트	타임라인 구성 테스트	☑																																			
	테스트	게시글/후기 기능 테스트	☑																																			
	테스트	프로필 및 팔로우/언팔 기능 테스트	☑																																			

기능 소개

검색 기능

검색어를 입력하면 해당 검색어와 관련된 게시글과 사용자가 조회되는 기능이다.

게시글은 검색어가 제목, 글 내용, 카테고리 중 하나라도 일치하는 내용이 있으면 검색된다.
사용자는 자기소개나 관심사에 일치하는 내용이 있으면 검색된다.

기능은 크게 키워드 검색과 해시태그 검색이 있다.
해시태그 검색은 개인 프로필 내에 있는 관심사에 링크를 걸어 해당 해시태그를 클릭 시 그 해시태그를 검색어로 검색된 결과가 화면에 출력된다.
키워드 검색은 키워드를 검색 창에 입력하면 검색된 결과가 화면에 출력된다.

trouble shooting

- 검색 시 게시글의 경우 제목 및 글 내용과 카테고리에서, 사용자는 자기소개와 관심사에서 검색결과가 중복된 결과를 걸러내지 못하는 오류가 발생했다.
→ 각각의 검색 결과의 배열들을 반복문을 통해 중복된 결과값을 뽑아내고 한 쪽의 배열에서 중복된 값을 제거하는 방식으로 문제를 해결했다.

회원가입 기능

회원가입 시 아이디(이메일), 비밀번호 닉네임, 생년월일 , 성별, 관심사를 기재하도록 되어있다.
아이디와 닉네임은 회원의 구분을 위해 중복 체크 기능을 넣었다.
'관심사'는 중복 체크가 가능하도록 했고 다른 테이블(`user_interest`)에 저장되도록 했다.
회원가입이 되면서 프로필도 자동으로 생성되도록 기능을 구현했다.

trouble shooting

- 회원가입 시 성별 선택(라디오)의 유효성 검사가 작동하지 않았다
→ 유효성 검사를 확인하는 스크립트에서 if 조건문을 라디오 값 == ""이라고 작성했던 것이 오류 발생의 원인이었다. 그래서 ""를 null로 바꿔줬다. 위에 조건문들은 다 ""로 작성하다보니 라디오도 이렇게 작성하게 된 것 같다.

- 동시에 user와 user_interest테이블에 저장할 때 user의 컬럼명 id를 userId로 가져오지 못하는 오류가 발생했다.
→ 회원가입 기능에서 객체로 가져오던 데이터를 파라미터로 가져오는 걸로 변경해 오류 해결했다.
- 관심사를 다중체크해도 user_interest 테이블에 하나의 값만 전달하는 문제가 발생했다.
→ 회원가입 기능을 구현한 코드에서 userInterest 파라미터를 전부 배열 객체로 바꾼 뒤 insert쿼리문에 foreach 반복문을 추가해 문제를 해결했다.

프로필 화면

프로필 화면에서는 사용자의 닉네임, 자기소개, 프로필 화면, 해시태그 형태로 된 관심사를 볼 수 있다. 또한 사용자가 주최한 모임 리스트와 사용자가 참여신청을 한 내역 리스트를 한번에 볼 수 있으며 주최 및 참여할 일정들도 캘린더에 표시되도록 화면을 구성했다,

팔로우/언팔로우가 가능하며 자신의 팔로워랑 자신이 팔로우하고 있는 사람의 수도 확인이 가능하다.

로그인 유저 본인만 프로필 수정이 가능하며 타인의 프로필에 들어갔을 경우 수정버튼은 나오지 않는다.

trouble shooting

- 닉네임 수정 시 userBO와 profileBO의 순환 참조 문제가 발생했다.
→ DB 설계를 다시했다. user테이블에 있던 nickname 컬럼을 profile 테이블로 바꿔서 문제를 해결했다.
- 프로필 수정 시 이미지를 변경하지 않고 수정하면 기존의 프로필 이미지가 삭제되는 오류를 발견했다.
→ profileBO의 updateProfile메소드에 조건문을 추가해 프로필이 수정되는 경우의 수를 4가지 정도로 나눠 분기문을 짰다. file 값이 null일 때와 새로 입력되었을 때로 나눠 기본값이 입력되도록 수정했다.

업데이트할 기능

- 팔로우 및 팔로워의 숫자를 누르면 팔로워 및 팔로워의 리스트를 볼 수 있는 기능을 구현할 예정이다.