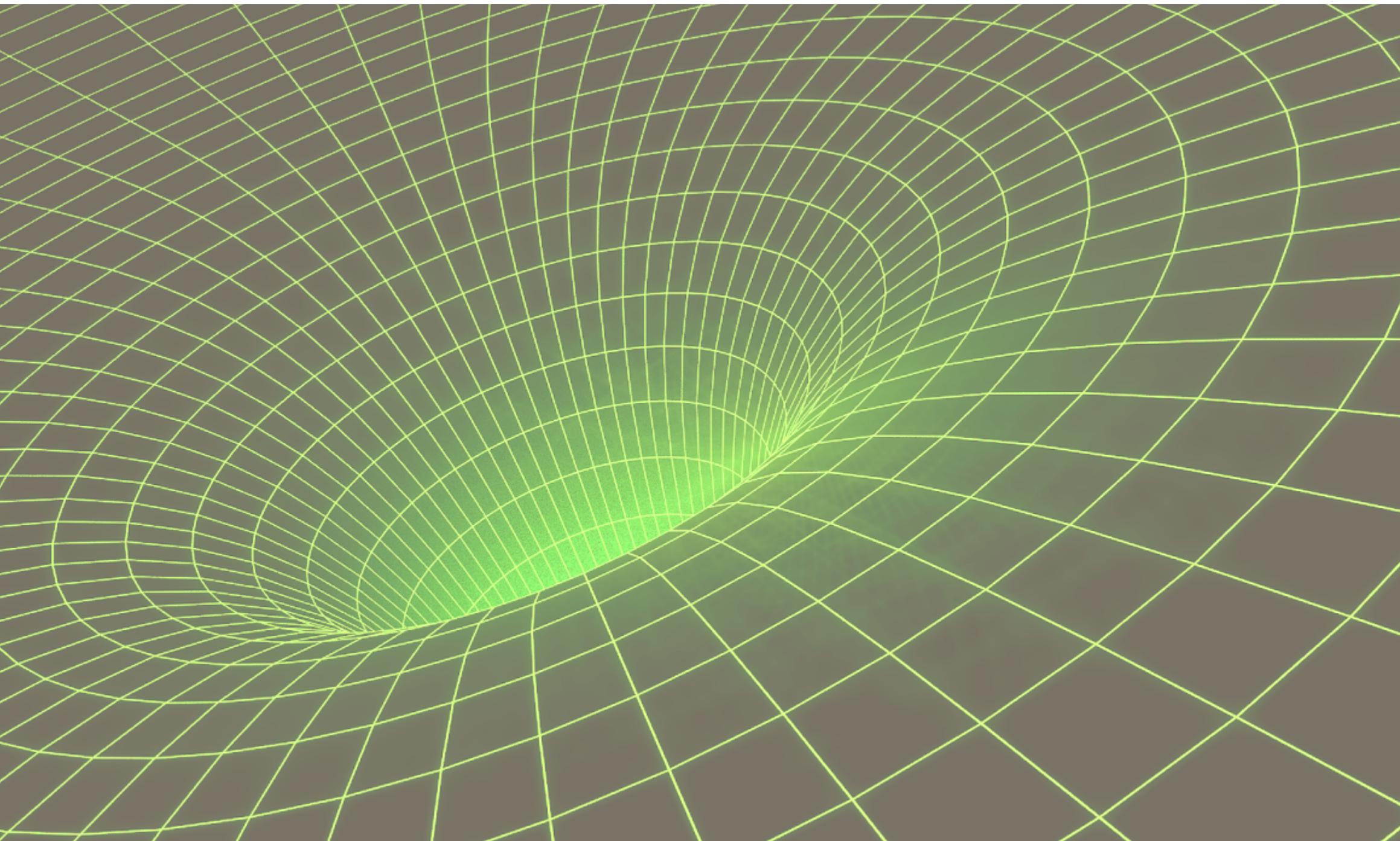


성실한 개발자



김진우 입니다.



김진우,
JINWOO KIM

Developer

kjinwoo2985@naver.com

010-7162-2985

<https://github.com/kjw7580>

김진우, JINWOOKIM



세부전공

강동대학교 컴퓨터 정보과 졸업

수행프로젝트

- 2021.08 - 2021-08 | 메모 프로젝트
메모기능이 포함된 메모장 프로젝트
- 2021.08 - 2021.09 | 마론달그램 프로젝트
SNS 기본 기능이 포함된 SNS 프로젝트
- 2021.09 - 2021.10 | KJW Health 프로젝트
로그인 기반, 소셜기능, 진단 / 기록실 기능을 포함한 건강관리 사이트 프로젝트

경력사항

- 2019년도 1학기 성적우수장학금
- 2019년도 2학기 성적면학장학금
- (주)블루젠틴 현장실습

개인프로젝트

- 2021.08 - 2021-08 | 메모 프로젝트
메모기능이 포함된 메모장 프로젝트
- 2021.08 - 2021.09 | 마론달그램 프로젝트
SNS 기본 기능이 포함된 SNS 프로젝트
- 2021.09 - 2021.10 | KJW Health 프로젝트
로그인 기반, 소셜기능, 진단 / 기록실 기능을 포함한 건강관리 사이트 프로젝트

자격증

- 네트워크 관리자 2급
- GTQ 1급
- IEQ 2급

Memo, 메모장 PROJECT

프로젝트 소개

- Memo PROJECT

Spring framework를 이용해 진행한 **첫번째** 프로젝트입니다. 간단한 메모 기능이 구현되어 있습니다.

로그인 기반의 프로젝트이고, 메모 작성 / 수정 / 삭제 / 다음페이지 / 이전페이지 기능이 포함된 프로젝트입니다.

Memo

로그인

아이디

비밀번호

로그인

[회원가입](#)

Copyright 2021. memo all rights reserved.

Memo

김진우 님 [로그아웃](#)

메모 리스트

No.	제목	시간
4	안녕하세요.	2021-08-24 18:37:13
3	메모 입력 기능 테스트!	2021-08-24 03:26:44
2	사진 올리기 테스트	2021-08-23 20:34:12

<이전

다음>

[글쓰기](#)

Copyright 2021. memo all rights reserved.

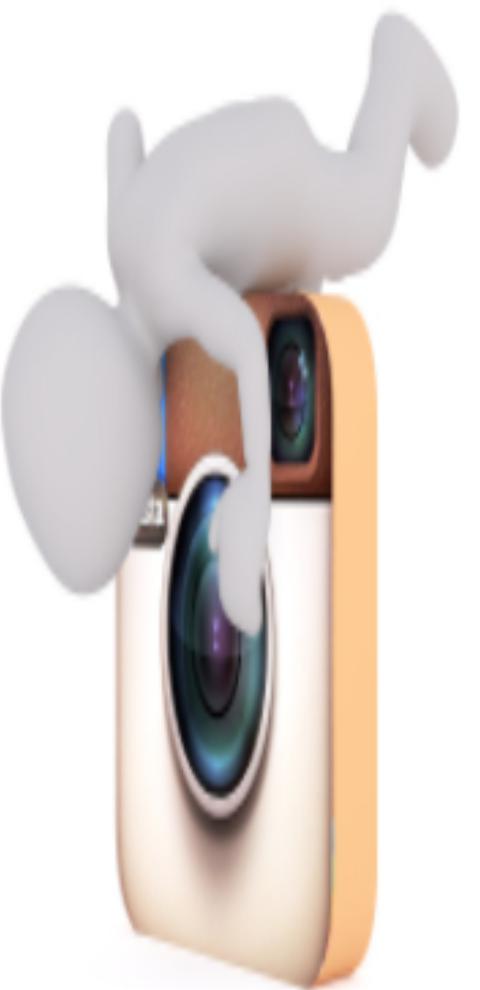
SNS PROJECT

프로젝트 소개

- SNS PROJECT

Spring framework를 이용해 진행한 **두번째** 프로젝트입니다. SNS 기본 기능이 구현되어 있습니다.

로그인 기반의 프로젝트이고, 글 업로드 / 삭제 / 댓글 / 좋아요 기능을 포함한 프로젝트입니다.



Marondalgram

전화번호, 사용자 이름 또는 이메일

비밀번호

로그인

[Marondalbook으로 로그인](#)

[비밀번호를 잊으셨나요?](#)
[비밀번호 찾기](#)

[계정이 없으신가요? **가입하기**](#)

kimjinwoo



좋아요 1개

kimjinwoo 오랜만에 기능 테스트입니다!!!

2021-09-14

댓글

kimjinwoo 오랜만에 댓글 써봅니다!!

댓글 달기... **게시**

KJW Health

건강관리 사이트 PROJECT

프로젝트 소개

KJW Health

메모장 / SNS 프로젝트를 마친 후 좀 더 웹 서비스 다운 프로젝트를 진행해보고 싶어 코로나 시국, 건강관리가 잘 안되고 있는 시점에 저만의 건강관리 사이트를 프로젝트로 진행해 보았습니다.

처음부터 프로젝트 **기획**, 데이터베이스 **설계**, 일정 설계, URL설계 부터 프로젝트 세팅, **기능 구현 및 테스트** 까지 직접 진행한 **개인 프로젝트** 입니다.

로그인 기반의 프로젝트이고, 진단하기, 진단결과 기록, 식단추천, 건강용품/식품 추천, 기록실, 건강 뉴스 등의 기능 정보 공유 게시판을 통한 **소셜기능**을 포함한 프로젝트입니다.

The screenshot displays two main pages of the KJW Health project:

- Home Page:** Features a navigation bar with Home, Search, and a login button. Below are sections for Health Products/Food (e.g., bicycle, resistance bands, protein bars), Diet Recipes (e.g., sweet potato, tomatoes, jump rope), and Health News (e.g., almond, fruit, vegetable soup).
- Diagnosis Result Page:** Shows AI self-diagnosis results, a list of users (131313, 121212, 111111, 101010, 999, 555, 444, 333, 222), and a summary of the user's health status.

Both pages include a watermark for "Copyright 2021. KJW Health all rights reserved."

KJW Health 건강관리 사이트 PROJECT

프로젝트 진행 후 느낀점

프로젝트 진행 후 느낀점

메모장 / SNS 프로젝트를 진행하면서 얻은 나를 어느정도의 Spring framework에 대한 이해를 바탕으로 진행한 개인 프로젝트입니다. 앞선 프로젝트에 비해 규모가 조금 더 크게 진행되었기에 기획 및 설계 과정에서 한번에 완벽하게 되지 않아 중간에 데이터베이스 설계를 한번 바꿔야하는 상황이 있었지만, 이과정을 통하여 설계뿐만 아니라 서비스의 흐름에 대해서도 더 잘 파악할 수 있었습니다. 데이터베이스 설계가 잘못 되었다는 것을 인지하고, 수정해 나가는 과정 속에서 전체적인 그림을 보는 능력을 키울 수 있었습니다.

이번 프로젝트를 진행하면서 웹 브라우저에서 Controller / BO / DAO / Mapper로 요청하고 다시 return 받는 과정까지의 흐름을 조금 더 명확하게 이해를 하게 된 계기가 되었습니다. 프로젝트 세팅부터 시작해서 초반에 조금 어려움도 있었지만, 에러가 발생하면 에러 메시지를 통해 코드 내용을 확인하며 혼자 해결해 보려고 노력했습니다. 혼자 해결이 안될 경우에는 구글 검색을 통해 에러내용을 찾아 해결하였습니다. 진행 초반에는 에러를 해결하는데 정말 많은 시간을 투자해야 하였습니다. 하지만 이 과정을 통해 성취감을 느끼고, 해결하는 과정속에서 Spring framework의 흐름을 파악하는데에도 도움을 많이 받았습니다. 혼자 에러를 해결해 나가는 과정이 반복되었고, 조금 더 복잡한 에러가 발생하여도, 해결해 나갈 수 있는 방법 그리고 능력을 얻을 수 있었습니다. 발생한 에러에 대해 생각하고 고민하는 과정이 개발능력을 향상 시킬 수 있는 중요한 과정이라는 것을 깨닫게 되었습니다. 그 후에도 해결과정에서 얻는 경험으로 발생한 에러들을 하나 하나 해결해 나갔던 것 같습니다.

또한 새로운 기능을 만들때, 영감을 얻기위해 검색을 통하여 여러 자료들을 찾아보면서, 제가 진행하던 구조 및 문법과 대부분 달랐던 경험이 많았습니다. 하지만 다른 구조 및 문법을 보고 흐름을 파악하고, 그 흐름을 제가 진행하던 구조 및 문법에 맞추어 바꿔서 적용을 많이 해보았습니다. 초반에는 애를 쓰고 해봤지만, 잘 적용이 되지 않았습니다. 계속 도전하고 노력을 통하여, 구조 및 문법을 바꾸어 적용하는 과정 또한 성공하였습니다.

마지막으로 진행과정을 하루 혹은 작업이 끝날때마다 Git을 통해 commit을 해두고, 발생하는 에러 및 중요한 포인트 위주로 commit을 해두어, 전체적인 진행과정을 확인하는데 도움을 받았습니다. 또한 Github와 연계된 Zenhub라는 툴을 이용해 기능별로 issue들을 등록하여, 기능마다 다른 issue에 commit함으로써, 기능별로 한눈에 어떻게 진행되었는지 확인할 수 있어서 편리하고 좋았습니다.

PROJECT NO.1

- KJW Health

Git hub 주소

[https://github.com/kjw7580/spring
_KJWHealth_0913](https://github.com/kjw7580/spring_KJWHealth_0913)

License

Tomcat [Apache License 2.0]
Mysql [GPLv2 or proprietary]
Spring framework [Apache License 2.0]
Mybatis [Apache License 2.0]
Bootstrap [MIT License]
JQuery [MIT License]
이미지 출처 : pixabay

기술 정보

KJW Health

기술 스택

Spring framework, JavaScript
Spring boot, JAVA, MySQL, AWS
JQuery, Bootstrap, HTML, CSS, Git
Github, Mybatis, JSTL

UI 기획



1. 코로나로 인해 평소보다 건강관리가 안되고 있는 시점에서 생각한 프로젝트 개발

2. 개인별로 건강관리를 할 수 있는 사이트 개발

3. 로그인 기반의 프로젝트 / 진단하기, 진단결과 기록, 식단추천, 건강용품/식품 추천, 기록실 등의 기능 개발

4. 정보 공유 게시판을 통한 소셜기능을 포함한 프로젝트

The screenshot displays a user interface for a health management project. At the top right, there is a logo for 'KJW Health' with a cross icon, followed by navigation links: 진단하기 (Diagnosis), 건강용품/식품 (Health Products/Food), 건강 정보 공유 (Health Information Sharing), 나의 건강 기록 (My Health Record), 회원가입 (Sign Up), and 로그인 (Login). Below the navigation, there are several sections: '나의 건강지표 차트' (My Health Indicators Chart) with a large circular chart area; '건강용품/식품' (Health Products/Food) with a placeholder '전체보기' (View All); '용품/식품 리스트' (Product/Food List) with a placeholder '전체보기'; '건강 정보 공유 게시판' (Health Information Sharing Board) with a placeholder '전체보기' and a list of users: kjw12, woomeu, jinwookim, kjw7580; and a '진단하기/결과' (Diagnosis/Result) section with a placeholder '전체보기'. A blue button at the bottom left says '+ 페이지 추가' (Add Page). A small note at the bottom right says '이 페이지의 최근 저장시간 : 2021.09.09'.

Ovenapp.io 서비스를 이용하여 전체적인 흐름을 잡을 수 있는 기획서 작성

작업역할 | UI 기획, DB 설계, URL 설계, 일정 설계, 기능 구

현 및 테스트

작업시기 | 2021.09 - 2021.10



기획 정보

[https://ovenapp.io/project/Jf2xG5QEngfmB07IG3vcTgfWIBkGGOeG
#VgL67](https://ovenapp.io/project/Jf2xG5QEngfmB07IG3vcTgfWIBkGGOeG#VgL67)

데이터베이스 설계



1. 코로나로 인해 평소보다 건강관리가 안되고 있는 시점에서 생각한 프로젝트 개발

2. 개인별로 건강관리를 할 수 있는 사이트 개발

3. 로그인 기반의 프로젝트 / 진단하기, 진단결과 기록, 식단추천, 건강용품/식품 추천, 기록실 등의 기능 개발

4. 정보 공유 게시판을 통한 소셜기능을 포함한 프로젝트

테이블 이름 : diagnosisResult

<code>id</code>	<code>int</code>	N	Y	primary key
<code>userId</code>	<code>int</code>	N	N	결과 나온 사용자 id (pk)
<code>selfDiagnosisContent</code>	<code>text</code>	N	N	AI 자가진단 결과
<code>dietName</code>	<code>varchar(32)</code>	N	N	개인별 식단 추천
<code>healthProductsName</code>	<code>varchar(32)</code>	N	N	개인별 건강용품/식품 추천
<code>dietImagePath</code>	<code>varchar(256)</code>	N	N	업로드된 이미지 경로
<code>healthProductsImagePath</code>	<code>varchar(256)</code>	N	N	업로드된 이미지 경로
<code>createdAt</code>	<code>timestamp</code>	N	N	생성 날짜
<code>updatedAt</code>	<code>timestamp</code>	N	N	수정 날짜

**Google 스프레드 시트를 통해 데이터 베이스 설계
테이블의 컬럼 / 컬럼 타입 / Null 여부
auto_increment 여부 / 설명 정보**

작업역할 | UI 기획, DB 설계, URL 설계, 일정 설계, 기능 구

현 및 테스트

작업시기 | 2021.09 - 2021.10

1

설계 정보

https://docs.google.com/spreadsheets/d/1_Q830KmiS-ayP8VYmQ40_vCBhxZ2f79swT_NdIE9BO4/edit#gid=0

URL 설계



1. 코로나로 인해 평소보다 건강관리가 안되고 있는 시점에서 생각한 프로젝트 개발

2. 개인별로 건강관리를 할 수 있는 사이트 개발

3. 로그인 기반의 프로젝트 / 진단하기, 진단결과 기록, 식단추천, 건강용품/식품 추천, 기록실 등의 기능 개발

4. 정보 공유 게시판을 통한 소셜기능을 포함한 프로젝트

View URL

제목	url	parameter	설명
메인페이지	/post/main		메인페이지 화면
진단하기 페이지	/post/diagnosis		진단하기 페이지 화면
진단결과 페이지	/post/diagnosis_result		진단결과 페이지 화면
건강용품/식품 페이지	/post/health_products		건강용품/식품 페이지 화면
나의 기록실 페이지	/post/record		나의 기록실 페이지 화면
식단 레시피 페이지	/post/recipe		식단 레시피 페이지 화면
게시글 목록	/post/list_view		게시글 목록 화면
게시글 쓰기	/post/create_view		게시글 쓰기 화면
게시글 상세 및 수정	/post/detail_view	id : 해당 글 id (pk)	게시글 상세 및 수정 화면
로그인	/user/signin_view		로그인 화면
회원가입	/user/signup_view		회원가입 화면

API URL

3) 식단 저장

URL : /post/diet/create

method : post

parameter	데이터 타입	NULL 여부	설명
파라미터 명	데이터 타입	NULL 여부	설명
maxBMI	int	N	BMI 최대 수치
minBMI	int	N	BMI 최소 수치
name	String	N	식단 이름
imagePath	MultipartFile	N	첨부 이미지 파일
recipe	String	N	레시피 내용

Google 스프레드 시트를 통해 URL 설계

View URL : url / parameter 정보

API URL : url / method / parameter / data type 정보

설계 정보

https://docs.google.com/spreadsheets/d/1_Q830KmiS-ayP8VYmQ40_vCBhxZ2f79swT_NdIE9BO4/edit#gid=0

일정 설계



1. 코로나로 인해 평소보다 건강관리가 안되고 있는 시점에서 생각한 프로젝트 개발
2. 개인별로 건강관리를 할 수 있는 사이트 개발
3. 로그인 기반의 프로젝트 / 진단하기, 진단결과 기록, 식단추천, 건강용품/식품 추천, 기록실 등의 기능 개발
4. 정보 공유 게시판을 통한 소셜기능을 포함한 프로젝트

작업역할 | UI 기획, DB 설계, URL 설계, 일정 설계, 기능 구현 및 테스트
작업시기 | 2021.09 - 2021.10

1

설계 정보

https://docs.google.com/spreadsheets/d/1_Q830KmiS-ayP8VYmQ40_vCBhxZ2f79swT_NdIE9BO4/edit#gid=0

프로젝트 일정								
날짜	내용	담당자	시작일	종료일	기간	특이사항	진행률	
2021/09/03	프로젝트 기획	김진우	2021-09-10	2021-09-10			100%	
2021/09/04	DB 설계	김진우	2021-09-11	2021-09-12			100%	
2021/09/05	URL 설계	김진우	2021-09-13	2021-09-13			100%	
2021/09/06	프로젝트 설계 / 세부	김진우	2021-09-14	2021-09-14			100%	
2021/09/07	디자인 / 구현 로그인 화면 디자인 / 구현	김진우	2021-09-15	2021-09-15			100%	
2021/09/08	로그인 화면 디자인 / 구현	김진우	2021-09-16	2021-09-16			100%	
2021/09/09	디자인 메인화면 화면 디자인	김진우	2021-09-17	2021-09-17			100%	
2021/09/10	간접흡연 / 식물 화면 디자인	김진우	2021-09-18	2021-09-18			100%	
2021/09/11	디자인 게시판 디자인	김진우	2021-09-19	2021-09-19			100%	
2021/09/12	디자인 식단 게시판 화면 디자인	김진우	2021-09-20	2021-09-20			100%	
2021/09/13	디자인 나의 기록화면 화면 디자인	김진우	2021-09-21	2021-09-21			100%	
2021/09/14	디자인 진단하기 화면 디자인	김진우	2021-09-22	2021-09-22			100%	
2021/09/15	디자인 진단결과 화면 디자인	김진우	2021-09-23	2021-09-23			100%	
2021/09/16	구현 / 버스터 게시판 게시글 작성 / 수정 / 삭제 기능 구현 및 버스터	김진우	2021-09-24	2021-09-25		Index 어려 해결	100%	
2021/09/17	구현 / 버스터 관리자 페이지 화면 구현 및 버스터	김진우	2021-09-24	2021-09-25			100%	
2021/09/18	구현 / 버스터 간접흡연 화면 구현 및 버스터	김진우	2021-09-26	2021-09-27			100%	
2021/09/19	구현 / 버스터 식단 화면 구현 및 버스터	김진우	2021-09-28	2021-09-29			100%	
2021/09/20	구현 / 버스터 자기진단 결과 화면 구현 및 버스터	김진우	2021-09-28	2021-09-29			100%	
2021/09/21	구현 / 버스터 진단하기 화면 구현 및 버스터	김진우	2021-09-30	2021-10-01			100%	
2021/09/22	구현 / 버스터 진단결과 화면 구현 및 버스터	김진우	2021-10-02	2021-10-03			100%	
2021/09/23	구현 / 버스터 나의 기록화면 기록하기 기능 구현 및 버스터	김진우	2021-10-04	2021-10-05			100%	
2021/09/24	구현 / 버스터 메인페이지 화면 구현 및 버스터	김진우	2021-10-06	2021-10-07			100%	
2021/09/25	구현 / 버스터 로그인 화면 디자인 및 구현	김진우	2021-10-07	2021-10-08			100%	
version-1								
2021/09/26	구현 / 버스터 로그인 화면 / 삭제 기능 구현 및 버스터	김진우					100%	
2021/09/27	구현 / 버스터 기록할 자세히 보기 및 삭제하기 / 삭제 기능 구현 및 버스터	김진우					100%	
2021/09/28	구현 / 버스터 게시판 key를 검색하기 / 내가 쓴 게시글 View 구현 및 버스터	김진우					100%	
2021/09/29	구현 / 버스터 게시판 글쓰기 기능 구현 및 버스터	김진우					100%	
2021/09/30	구현 / 버스터 마이페이지 화면 구현 및 버스터	김진우					100%	
2021/10/01	구현 / 버스터 회시피 홍보링크로 홍보하기 기능 구현 및 버스터	김진우					100%	
2021/10/02	구현 / 버스터 간접흡연 홍보글 홍보하기 기능 구현 및 버스터	김진우					100%	
2021/10/03	구현 / 버스터 기록을 추천하는 식단 / 간접흡연식단을 찾고 있는 View 구현 및 버스터	김진우					100%	
2021/10/04	구현 / 버스터 기록을 기록한 친구들과 리스트로 확인하기 기능 구현 및 버스터	김진우				userlist 라이브 구현 성공	100%	
2021/10/05	구현 / 버스터 내가 쓴 게시글 View userid로 확인하기 기능 구현 및 버스터	김진우					100%	
2021/10/06	구현 / 버스터 게시판 key를 검색할 때 대량 데이터가 있을 때 구현 및 버스터	김진우					100%	
2021/10/07	구현 / 버스터 전체 페이지 Markus View 디자인 결정 및 구현 및 버스터	김진우					100%	
version-2								
개발 일정 기록								
개발 완료 기록								
개발 완료 기록								
개발 완료 기록								
후기 수렴할 수 있습니다!								

**Google 스프레드 시트를 통해 일정 설계
화면 Markup View / 기능 구현 / 테스트의 일정을 계획
계획보다 뒤쳐지지 않도록 노력
최대한 계획보다 빠르게 진행**

Git / Github Zenhub



1. 코로나로 인해 평소보다 건강관리가 안되고 있는 시점에서 생각한 프로젝트 개발

2. 개인별로 건강관리를 할 수 있는 사이트 개발

3. 로그인 기반의 프로젝트 / 진단하기, 진단결과 기록, 식단추천, 건강용품/식품 추천, 기록실 등의 기능 개발

4. 정보 공유 게시판을 통한 소셜기능을 포함한 프로젝트

작업역할 | UI 기획, DB 설계, URL 설계, 일정 설계, 기능 구현 및 테스트

작업시기 | 2021.09 - 2021.10

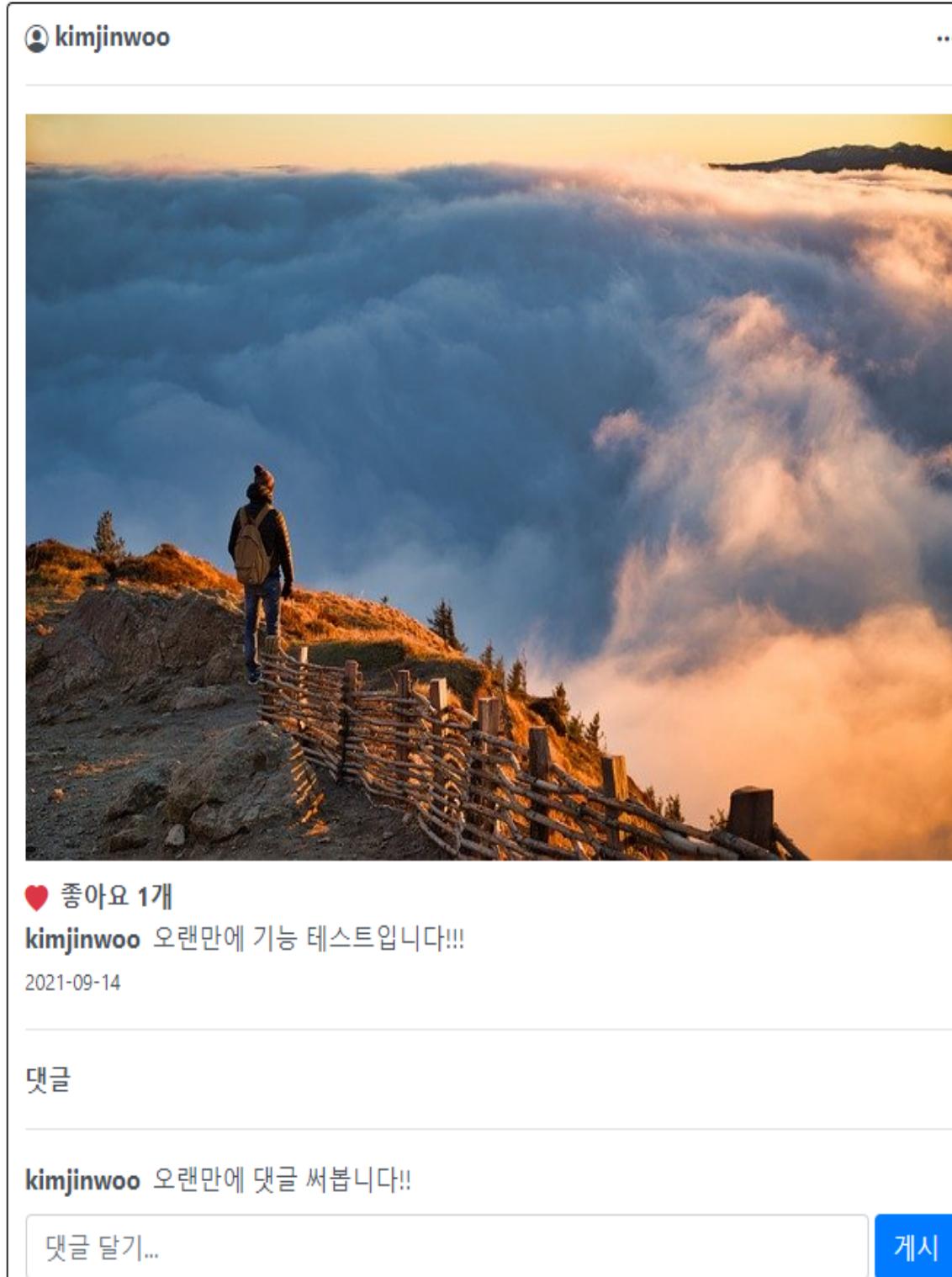
The screenshot shows the Zenhub interface. On the left, a sidebar lists various issues under a 'workspace' named 'WORKSPACE'. The 'History' tab is selected, showing a detailed log of commits. The commit log includes timestamp, author, commit ID, and a brief description of the changes made.

날짜	작성자	커밋 ID
10 10 2021 22:39	DESKTOP-PDEKLC	b285df1
10 10 2021 12:30	DESKTOP-PDEKLC	2a55f2e
10 10 2021 11:54	DESKTOP-PDEKLC	f33080f
10 10 2021 11:15	DESKTOP-PDEKLC	93752d5
10 10 2021 11:02	DESKTOP-PDEKLC	b232079
10 10 2021 10:57	DESKTOP-PDEKLC	a948692
10 10 2021 10:54	DESKTOP-PDEKLC	4345dfa
10 10 2021 0:54	DESKTOP-PDEKLC	b80e427
9 10 2021 23:53	DESKTOP-PDEKLC	3efc48b
9 10 2021 23:47	DESKTOP-PDEKLC	c4b9094
8 10 2021 21:08	DESKTOP-PDEKLC	3c00295
7 10 2021 23:55	DESKTOP-PDEKLC	280a51e
6 10 2021 20:24	DESKTOP-PDEKLC	34f0e1d
6 10 2021 0:45	DESKTOP-PDEKLC	cffbf07
5 10 2021 9:30	DESKTOP-PDEKLC	fa46bbf
1 10 2021 9:26	DESKTOP-PDEKLC	43c01d6
26 9 2021 18:47	DESKTOP-PDEKLC	24b4802
22 9 2021 16:15	DESKTOP-PDEKLC	e85fb63
22 9 2021 16:14	DESKTOP-PDEKLC	e88c181
22 9 2021 10:03	DESKTOP-PDEKLC	1c00fc5
22 9 2021 10:02	DESKTOP-PDEKLC	5c77ba5
22 9 2021 10:00	DESKTOP-PDEKLC	3c2ca24
16 9 2021 16:45	DESKTOP-PDEKLC	11a8af9

Sourcetree를 통해 하루 혹은 작업이 끝날때마다 commit. 발생하는 에러 및 중요한 포인트 위주로 commit. Github와 연계된 Zenhub를 이용해 기능별로 issue들을 등록 / 기능마다 다른 issue에 commit.

Github 주소
<https://github.com/kjw7580>

1 Memo / SNS PROJECT 좋아요 기능



Introduce

메모장 / SNS 프로젝트의 모든 기능에 아쉬운점을 보완하고 새로운 기능들을 추가하여 KJW Health 프로젝트를 진행하였기에, KJW Health에는 사용하지 않은 좋아요 기능 설명하도록 하겠습니다.
KJW Health에는 사용하지 않은 좋아요 기능 설명하도록 하겠습니다.

- SNS 좋아요 기능

유저마다 게시글에 좋아요 버튼을 눌렀을 때 좋아요 수가 올라가고 한번 더 클릭했을 때 좋아요 취소가 되는 기능입니다.

Solution

어떤 유저가 어떤 포스트에 좋아요를 눌렀는지 정보가 필요하기 때문에

INSERT 하는 과정에서 `postId` 와 `userId`를 INSERT 해줍니다.

SELECT를 하는 과정에서도 WHERE `postId` = #{postId} AND `userId` = #{userId} 조건을 걸어서
SELECT를 해주게 됩니다. DELETE 또한 같은 조건을 걸어줍니다.

return을 받아서 서버에서 int count를 통하여 if(count >= 1) => return true; else => return false 조건을 통해 좋아요를 했는지 확인합니다.

그 후 좋아요 상태가 되면 true, 좋아요 취소상태가 되면 false의 조건을 걸어줍니다. 그 후 Controller에서 데이터를 저장해줍니다.

좋아요 개수는 postId별로 데이터를 받아올 수 있도록 쿼리 작성할 때 WHERE `postId` = #{postId} 조건을 걸어 return 받아서 똑같이 Controller에서 데이터를 저장 해줍니다.

View 화면에서 넘겨받은 postId별 좋아요 개수를 적용해주도록 구현하였습니다.

Information

Memo / SNS 프로젝트 전체적인 설명 / 기능 Demo GIF / 코드 소스
는 Github를 통하여 보실 수 있습니다!

Memo Git hub 주소

https://github.com/kjw7580/spring_memo_0816

Marondalgram Git hub 주소

https://github.com/kjw7580/spring_marondalgram_0816

KJW Health PROJECT 회원 가입 / 로그인



아이디
비밀번호
비밀번호 확인
이름
이메일 주소

가입

계정이 있으신가요? [로그인](#)

Copyright 2021. KJW Health all rights reserved.



전화번호, 사용자 이름 또는 이메일
비밀번호

로그인

[비밀번호를 잊으셨나요?](#)
[비밀번호 찾기](#)

[계정이 없으신가요?](#) [가입하기](#)

Copyright 2021. KJW Health all rights reserved.

Introduce

-회원가입

- 전체적인 유효성 검사를 통한 회원가입 구조입니다.
- 중복확인을 통한 중복된 ID 가입을 방지합니다.
- 비밀번호 확인을 통해 비밀번호가 일치해야만 회원가입이 진행됩니다.

-로그인

- 로그인 또한 유효성 검사를 통해 로그인이 가능합니다.
- 만들어진 계정은 데이터 베이스에 암호화되어 저장됩니다.

Solution

Sign_up INSERT 과정을 통하여 회원가입이 완료 됩니다.

JQuery Click 이벤트를 통한 전체적인 유효성 검사가 이루어 집니다.

Boolean을 통해 중복확인 과정이 이루어집니다.

가입완료된 계정 Password는 암호화된 Password로 저장됩니다.

MD5를 통한 비밀번호 암호화 / Integer.toHexString를 통해 암호화된 비밀번호로 데이터베이스에 저장됩니다.

로그인이 되었을 때 session.setAttribute를 통해 userId / userLoginId / userName 세팅합니다.

세션에서 userId / userLoginId / userName 정보를 얻을 수 있게 됩니다.

로그아웃이 되었을 때 removeAttribute를 통해 세팅된 정보들을 remove 해줍니다.

Plan

보안적인 장치나 설정이 되어 있지 않습니다.

회원가입에는 개인 정보가 저장되기 때문에 보안적인 문제가 중요하다고 생각합니다.

다음에 기회가 된다면 보안 설정 / 보안관련 정보를 얻어서 보안을 강화하여 회원가입 / 로그인 기능을 만들어 보고 싶습니다.

Spring에서 제공하는 Spring security를 적용하여 보안에 신경써서 더 안전한 로그인 처리를 할 수 있도록 구현해보고 싶습니다.

2 KJW Health PROJECT 진단하기

진단하기

— KJW Health Diagnosis

[홈으로 돌아가기](#)

나이

키

몸무게

[입력하기](#)

당신의 BMI 지수는?

자가진단

— KJW Health Self Diagnosis

Introduce

-입력하기

메인 페이지 메뉴 -> 진단하기 클릭시 나타나는 화면입니다.

개인의 나이 / 키 / 몸무게를 입력받아서 BMI Table에 저장이 됩니다.

-진단하기

가장 최신에 저장된 정보를 불러와 BMI 수치를 구하게 됩니다.

구해진 BMI 수치를 기준으로 개인별 자가진단 / 식단 / 건강 용품식품을 추천 받게 됩니다.

Solution

BMI INSERT 과정을 통하여 정보가 저장됩니다.

```
SELECT * FROM `BMI` ORDER BY `id` DESC LIMIT 1
```

DESC LIMIT 1 조건을 걸어서 가장 최신의 정보(사용자가 입력한 정보)를 받아오게 됩니다.

받아온 정보를 서버에서 BMI 구하는 공식에 대입하여 BMI값을 구해서 넘겨주게 됩니다.

넘겨받은 BMI를 화면에 innerHTML를 통해 보여지게 됩니다.

- BMI 수치를 구하는 과정에서 형변환 error 발생

minBMI / maxBMI는 int형 인데, String으로 받았기 때문에 **try catch(NumberFormatException)**를 통해

예외 처리를 해준 후 error 해결

- 진단하기 페이지에서 계산된 BMI수치가 보여질 때 double형태로 보여지게 구현 필요

BMI수치를 소수점 형태로 전달 되어야 하기 때문에 **double 형태로 계산**하여 **Double.toString** 통해 문자열로 만들어서 넘겨주는 방식으로 해결

Plan

설문 기능을 추가하여 설문을 통해서 점수를 계산하여 예를 들어,

나의 건강점수 같은 기능을 진단결과 화면에 보여질 수 있게 만들어 보려고 합니다.

설문 질문에 체크한 갯수를 통하여 점수를 계산하는 방식으로 생각 중 입니다.

3 KJW Health PROJECT 진단결과

K 진단결과

[홈으로 돌아가기](#)

Introduce

-진단결과

구해진 BMI 수치를 기준으로 각각의 자가진단 / 식단 / 건강용품식품 테이블에 저장된 minBMI / maxBMI를 통하여 범위를 지정하여 자가진단 / 식단 / 건강용품식품을 추천 받게 됩니다.

K AI 자가진단 결과

[기록하기](#)

첫번째 자가진단입니다!!!
두번째 자가진단입니다!!!

K 개인별 식단 추천



아몬드

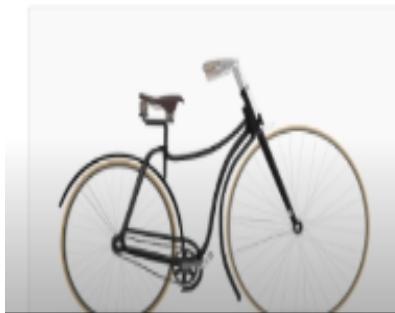


과일



야채 수프

K 개인별 건강용품/식품 추천



Solution

처음에 화면에 보여지는 자료들을 어떻게 넘겨줄 수 있을지 고민을 많이 했습니다. 화면에 보여지는 이미지들과 이름들을 일일이 다 받아오게 데이터베이스를 설계했습니다. **설계가 잘못되었다는 것을 인지하고** 이미지들과 이름들을 일일이 받아오는 것이 아닌 id 자체를 받아올 수 있게 **설계를 다시 하였습니다.**

id정보를 리스트로 넘겨줄 수 있는 방법을 찾아보다가 이 방법 저 방법 적용해봤지만, 해결이 되지 않아 **문자열로 넘겨주는 방법**을 선택했습니다.

ajax 통신을 이용하여 화면에 뿌려지는 각각의 자가진단 / 식단 / 용품식품들의 id들을 반복문을 통하여 id를 구한 후 나름의 규칙, 저는 쉼표를 통해 문자열로 만들어서 서버로 넘겨주게 됩니다.

서버에서는 넘겨받은 문자열을 반복문을 통해서 **쉼표를 기준으로 split**을 해주고 **Integer.parseInt**를 해주어 Integer형으로 바꾸어 준 후에 **리스트에 저장**하여 **model.addAttribute**를 통하여 id값을 넘겨주게 됩니다.

넘겨받은 id 정보를 통하여 기록실에 저장이 될 수 있게 구현되어 있습니다.

Plan

진단결과 페이지에 진단결과 / 식단 / 용품식품 추천 말고 또 다른 추가 항목을 추가하여 예를들어) 운동법 / 칼로리 표시 등등.. 더 다양하게 서비스를 이용할 수 있도록 구현해보고 싶습니다.

쉼표로 구분하여 넘겨준 데이터들은 나중에 **jackson 라이브러리**를 활용하여 **json 형태**로 변환해서 **안정성과 확장성**이 확보 될 수 있도록 구현해보도록 하겠습니다.

4 KJW Health PROJECT 게시판 게시글, 댓글쓰기 / 수정 / 게시글, 댓글삭제 기능

제목	글쓰기 기능 테스트!!!
내용	글쓰기 기능 테스트입니다!!!
댓글	김진우 댓글 기능 테스트입니다!!
	<button>삭제</button> <button>게시</button> <button>수정</button>
목록으로	삭제

제목	제목을 입력하세요.
내용	내용을 입력하세요.
파일 선택	선택된 파일 없음
목록으로	저장

Introduce

- 게시글, 댓글쓰기

글쓰기 페이지에서 제목 / 내용을 입력한 후 게시를 누르게 되면 게시글이 작성됩니다 / 댓글내용을 입력한 후 게시 버튼을 누르면 댓글이 작성됩니다.

- 수정

수정 하고 싶은 제목과 내용을 입력한 후 수정 버튼을 누르게 되면 게시글 정보가 수정됩니다.

- 삭제

원하는 게시글 혹은 댓글의 삭제 버튼을 누르게 되면 게시글 / 댓글이 삭제됩니다.

내가 게시한 글, 댓글만 삭제 및 수정할 수 있도록 구현되어 있습니다.

Solution

INSERT 쿼리를 통해 게시글 정보가 데이터베이스에 저장됩니다. / INSERT 쿼리를 통해 댓글 정보가 데이터베이스에 저장됩니다. Update SET '제목', '내용' 쿼리를 통해 정보가 수정됩니다.

DELETE WHERE `id` AND `userId` 조건을 걸어서 조건을 기준으로 게시글과 댓글이 삭제됩니다.

게시판 특성상 제가 게시한 글 / 댓글만 수정 혹은 삭제가 가능하게 해야하므로, View화면쪽에서 보여질 때

<c:if test="\${게시글 작성한 userId eq 세션의 userId }"> 의 조건을 걸어서 게시글 작성한 userId와 세션의 userId정보가 일치할 때 수정 및 삭제버튼이 보이도록 구현되어 있습니다.

<c:if test="\${댓글 작성한 userId eq 세션의 userId }"> 댓글도 같은 구조입니다.

댓글 삭제 기능을 구현할 때 삭제 버튼에 **data-comment-id="\${comment.id }**의 아이디를 부여한 후

var postId = \$(this).data("post-id"); 로 postId 정보를 꺼낸 후 **ajax 통신을 통하여 postId를 서버로 넘겨주**는 방식으로 구현 했습니다.

Plan

댓글 수정 기능 추가

게시글 추천 기능 추가하여 구현해보고 싶습니다.

5 KJW Health PROJECT 게시판 리스트 내가 쓴 게시글 리스트 / 페이징 기능

No.	제목	작성자	시간
18	관리자 두 번째 공지입니다!!!!!!	2	2021-10-08 11:14:39
17	관리자 공지입니다!!!	2	2021-10-08 11:14:20
12	후기 알려드린니다~!!	1	2021-09-25 06:52:05
11	이거 아시는분~?!	1	2021-09-25 06:51:31
10	이거 어떤가요? 의견좀요!!	1	2021-09-25 06:51:12
9	게시판 이용 팁 알려드립니다!!!	1	2021-09-25 06:50:54
8	게시판 자주 자주 이용합시다~	1	2021-09-25 06:50:28
7	게시판 활성화!!!	1	2021-09-25 06:50:11
6	추천 받습니다~!	1	2021-09-25 06:49:56
5	식당 추천해주세요!!!	1	2021-09-16 17:31:56

글쓰기

1 2 3

Copyright 2021. KJW Health all rights reserved.

No.	제목	작성자	시간
	글쓰기 기능 테스트!!!	1	2021-10-13 23:04:08
	121212	1	2021-10-10 00:00:22
	111111	1	2021-10-09 21:04:24
	101010	1	2021-10-09 21:04:18
	999	1	2021-10-09 21:04:10
	555	1	2021-10-09 20:31:27
	444	1	2021-10-09 20:31:21
	333	1	2021-10-09 20:31:13
	222	1	2021-10-09 20:31:08
	111	1	2021-10-09 20:31:01

1 2

Copyright 2021. KJW Health all rights reserved.

Introduce

- 게시판 리스트 화면

모든 유저들이 작성한 모든 게시물 리스트가 화면에 보여집니다.

정보가 많아지면 리스트가 끝없이 내려가기 때문에 페이징 기능을 추가하여 구현되어 있습니다.

- 내가 쓴 게시글 리스트

모든 게시물 리스트가 화면에 보여지기 때문에 내가 어떤 게시물을 작성했었는지 쉽게 확인할 수 있도록 내가 쓴 게시글 리스트 화면을 추가로 구성하였습니다.

내가 쓴 게시글 리스트에서 userId별로 작성된 게시글을 페이징 기능을 추가하여 구현하였습니다.

Solution

게시판에 들어가게되면 게시글의 모든 정보를 불러오게 됩니다.

하지만 프로젝트를 진행하면서 게시글을 많이 써보니, 화면이 끝없이 내려가 불편함을 겪었고, 페이징 기능은 필수적으로 구현을 해야한다고 생각했습니다.

처음 쿼리를 작성할 때 LIMIT #{page}, 10 조건을 통하여 한페이지에 10개의 게시글의 정보를 가져오는 것을 확인하였습니다. 하지만 한 페이지에 보여지는 데이터가 10개가 아니면 LIMIT 부분의 마지막에 입력된 10의 숫자를 변경해야하는 **문제가 발생**했고, 매번 원하는 페이지를 처리할 때마다 **계산이 필요**하다는 문제가 있었습니다.

우선 페이지 번수 / 한페이지에 보여지는 데이터 갯수의 **변수를 지정**해서 model package에 **Criteria Class**를 만들어 주었습니다. 그리고 totalCount(총 갯수) / 시작페이지 / 끝페이지 **변수를 지정**해서 마찬가지로 model package에 **PageMaker Class**를 만들어 주었습니다. 그리고 calcDate 계산식을 만들어 준 후 게시글의 **전체 갯수가 결정**되면 calcData **메소드를 호출하여 계산이 실행**되게 해주었습니다. 그리고 page 값이 **음수가 될 수 없도록 설정**을 해주었습니다. 이 세팅을 마친 후 LIMIT #{page}, 10 아닌 LIMIT #{시작페이지}, #{Criteria Class}에 지정한 한페이지에 보여지는 데이터 갯수의 변수}로 수정하여 **새로 쿼리를 작성**한 후 return 받아서 model.addAttribute에 세팅 후 넘겨줍니다. 그리고 **전체 게시글 갯수**를 구하는 과정이 필요했습니다. 전체 게시글 갯수를 가져오는 쿼리 COUNT(`id`)를 작성하여 return 받아 **totalCount(총 갯수)**를 PageMaker Class에 만들어 놓은 **setter**에 **세팅**을 해줍니다. PageMaker 또한 model.addAttribute에 세팅 후 넘겨줍니다.

JSP에서 넘겨 받은 정보를 이용해 paging을 위한 JSP파일을 하나 더 만들어 준 후 원하는 페이지 버튼을 눌렀을때 ?page=\${idx}라는 **조건을 걸어**주어 페이징 세팅을 해줍니다. 다음은 내가 쓴 게시글 화면을 구현 중 userId를 기준으로 받아와야 하기 때문에 전체 리스트 페이징했을 때와 같은 쿼리를 작성한 후 WHERE 조건에 `userId`=#{userId} 조건을 걸어 userId별로 가져올 수 있게 return 받아 실행해 봤지만 **error가 발생**했습니다. View쪽에 userId가 일치할 때의 조건을 걸어 실행해보았습니다. 실행은 되었지만, 다른 작성자가 작성한 게시글은 그대로 공백처리가 되어 화면에 보여졌습니다. 고민을 하다가 Criteria Class에 **userId 세팅**이 안되어 있다는 것을 **인지**하고, getter/setter를 만들어 준 후 서버에 **setUserId**를 해주어 userId별로 페이징을 할 수 있도록 **해결**하였습니다.

6 KJW Health PROJECT 검색기능 / 검색 결과에 대한 페이징

The screenshot shows a search interface with a search bar containing '검색' (Search) and dropdown menus for '제목' (Title), '내용' (Content), '게시날짜' (Post Date), '제목/내용' (Title/Content), '제목/게시날짜' (Title/Post Date), '내용/게시날짜' (Content/Post Date), and '전체' (All). Below the search bar is a link '내가 쓴 게시글 보러가기' (View my posts). A table lists search results:

No.	제목	작성자	시간
12	후기 알려드립니다~!!!	1	2021-09-25 06:52:05

A blue button labeled '글쓰기' (Write a post) is visible on the right side of the table.

Introduce

- 검색 기능

key 별로 기능을 할 수 있는 기능입니다.

원하는 key(제목, 내용, 게시날짜, 제목/내용, 제목/게시날짜, 내용/게시날짜, 전체)를 선택하여 해당하는 조건에 맞추어 검색이 가능 합니다.

- 검색 결과에 대한 페이징

원하는 조건으로 검색을 했을 경우, 그 검색 결과의 게시글 갯수에 따른 페이징 기능을 구현하였습니다.

Solution

첫번째로 model package -> Post Class에 key(검색할때 선택된 값) / word(검색할 내용)를 지정해준 후 getter/setter를 만들어 주었습니다. 그 후 View에서 select option을 통하여 option마다 value를 만들어 준 후, key=value일때의 조건을 걸어주었습니다.

원하는 option만큼 option을 만들어 준 후, 검색에 대한 쿼리를 작성하였습니다.

<sql id=""> 사용법을 통해 검색어가 존재 하는 경우에 검색을 하고, option마다 해당되는 조건을 걸어주어 작성하여 작성한 쿼리의 id를 지정한 후, 이 쿼리를 수행하고 싶은 쿼리에 <include refid="">를 통해 include 해주었습니다. 그 후 서버에 key, word 를 setter를 통해 세팅을 해준 후 실행해보았습니다. 검색결과에 대한 결과는 잘 나왔지만, 검색결과에 대한 페이징이 실행되지 않았습니다. include가 되어있는 쿼리가 전체 데이터 페이징을 가져오는 쿼리였기 때문에 검색을 했을때 전체 데이터에 대한 페이징이 되는 것이라 판단을 하였습니다.

그래서 조건마다 쿼리를 따로 만들어 주는 방법을 선택했습니다.

예를들어)key가 제목일때, key가 내용일때 등등.. key별로 따로 쿼리를 작성하여 return 받아 서버에서 조건문을 통해서 조건마다 그 해당되는 조건의 쿼리를 가져올 수 있도록 세팅을 해주어 검색결과에 대한 페이징 문제를 해결하였습니다.

Plan

조회수 기능 추가

조회수 기능을 통한 인기를 설정 기능

게시판 Category 추가

7 KJW Health PROJECT 나의 기록실 / 페이지

기록된 Date : 2021-10-13 23:15:59 삭제

[진단결과 보러가기](#)

[레시피 보러가기](#)

[건강용품/식품 보러가기](#)

기록된 Date : 2021-10-10 11:48:51 삭제

[진단결과 보러가기](#)

[레시피 보러가기](#)

[건강용품/식품 보러가기](#)

기록된 Date : 2021-10-08 15:24:06 삭제

[진단결과 보러가기](#)

[레시피 보러가기](#)

[건강용품/식품 보러가기](#)

1 2 3

Introduce

- 나의 기록실

진단결과 페이지에서 기록하기 버튼을 눌렀을 때 기록실에 저장됩니다.

기록실의 저장된 기록은 기록했을 당시 정보들이 저장되고,

보러가기를 눌렀을 때 기록된 정보들을 다시 볼 수 있도록 설정 해두었습니다.

기록들이 쌓이게 되면 복잡해질 수 있으므로 역시 페이징기능 / 삭제기능을 추가하였습니다.

Solution

진단결과 페이지에서 기록하기 버튼을 누르면 진단결과에 보이는 정보들이 저장되게 됩니다. 위에 설명되어 있지만 ajax 통신으로 넘겨받은 데이터를 같은 방식(반복문을 통하여 쉼표를 기준으로 split해준 후 Integer.parseInt로 형변환 후 리스트에 저장)으로 원하는 데이터들을 model.addAttribute를 통해 넘겨주게 됩니다.

진단결과 보러가기에서는 자가진단 / 식단 / 용품식품 데이터들을, 레시피는 식단 데이터를, 용품식품은 용품식품 데이터를 View화면에 넘겨줍니다. 각각의 페이지의 상세페이지 View를 새로 만들어 줍니다.

기록실 화면 View에 각각의 버튼을 생성해주고 버튼에 링크를 걸어줄 때 반복문을 통하여 진단결과 데이터의 아이디를 빼낸 후 ?id=\${diagnosisResult(진단결과).id } 조건을 걸어주어 id별로 상세페이지로 넘어갈 수 있도록 세팅해 줍니다.

상세페이지 View에서도 넘겨받은 정보를 사용하여 반복문을 통해 정보를 빼내어 화면에 뿌려지게 구현하였습니다.

기록실 페이지 또한 내가 기록한 기록들만 보여야 하기 때문에 userId별로 페이징을 해두었습니다.

Plan

기록실에 날마다 그날의 컨디션이라던지, 건강상태 변화, 진단결과에 대한 간단한 메모를 할 수 있는 기능을 구현해보고 싶습니다.

8 KJW Health PROJECT 건강뉴스 / 상세페이지 / 페이지

K 건강 뉴스

전체보기



다음주 출범하는 KB헬스케어... '파격 건강서비스' 예고



'대들보 척추' 건강 지키는 '연령별 스트레칭'



국립재활원 장애인건강검진센터 개원...유형·중증도 별 특화

Introduce

- 건강 뉴스

유용한 정보를 얻을 수 있는 기사들을 볼 수 있는 페이지입니다.

- 상세 페이지

기사마다 상세페이지를 볼 수 있습니다. 자세히 보기 를 누르게 되면 전체 기사 내용을 보실 수 있습니다.

- 메인 뉴스 페이지

메인 페이지에 최신 올라온 3개의 기사를 접하실 수 있으며, 제목을 보고 관심이 있으면 들어가 볼 수 있게 설정해두었습니다.

건강 뉴스 추천
— KJW Health News

목록으로

K KJW Health

건강 뉴스

— KJW Health News

홈으로 돌아가기

다음주 출범하는 KB헬스케어... '파격 건강서비스' 예고



헬스케어 서비스는 이용자의 질병에 대해 발생시 치료는 물론 예방과 관리, 건강관리·증진 서비스 등을 종합적으로 제공할 수 있는 것이다. 보험업계는 건강관리 서비스와 건강증진형 보험상품 등의 방식으로 헬스케어 서비스를 제공하고 있다. 보험사가 헬스케어 자회사의 플랫폼에서 모인 건강데이터를 마이데이터 사업과 결합하면 새로운 건강증진형 상품이나 서비스 출시로 연결될 수 있다. KB헬스케어가 제공하는 주요 서비스는 모바일 애플리케이션(앱)을 통한 디지털 건강관리 서비스(▲건강검진 정보 등 다양한 건강정보를 분석한 건강상태 정보 ▲고객별 건강상태 기반 건강목표 추천 ▲식단 데이터 분석서비스 등)와 고객의 건강관리 목표 달성을 지원하기 위해 외부 제휴업체와 연계한 건강관리 서비스(▲유전체 분석 서비스 ▲오픈·비디오 기반의 디지털 활동관리 프로그램 ▲ 만성질환자 건강관리 코칭 프로그램 ▲멘탈 관리 상담 프로그램 등)로 구성될 예정이다. 실제 KB손해보험은 지난달 24일 온라인 채널인 디아렉트 전용 앱에서 헬스케어형 특약과 결합한 서비스를 출시했다. KB손해보험이 디아렉트 전용 앱에서 공개한 '케이비-워크(KB-WALK)'는 이용자들의 걸음 수를 매일 체크하는 동시에 자동차보험의 '걸음수 할인특약'을 쉽게 가입할 수 있는 기능이다. KB손해보험 '걸음수 할인특약' 가입자가 보험 청약일 기준 90일 이내 하루 5000보 이상 달성한 날이 50일 이상인 경우 자동차보험료를 3% 할인 받을 수 있는 서비스다.

Solution

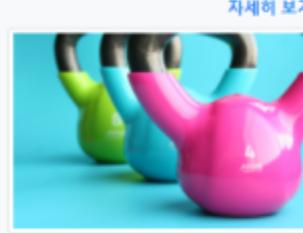
News 테이블의 정보들을 SELECT 받아오는 과정에서 WHERE `id` = #{id}의 조건을 걸어 아이디기준으로 받아와서 자세히 보기 를 누르게 되면 해당 기사의 전체 기사 내용을 볼 수 있게 구현되어 있습니다.

뉴스 리스트 화면에서 긴 제목이 그대로 다 노출이 되면 복잡해 보일 수 있으므로 `{fn:length() gt } / ${fn:substring()}`를 통하여 특정 글자 수가 넘어가면 ...으로 처리를 하였습니다.



다음주 출범하는 K...

자세히 보기



'대들보 척추' 건...

자세히 보기



국립재활원 장애인건...

자세히 보기



"보험금을 안 줘요...

자세히 보기



WSJ "애플, ..."

자세히 보기



대상웰라이프 뉴케어...

자세히 보기



건강보험료 36만원...

자세히 보기



"건강보험료, 소득...

자세히 보기

1 2

Copyright 2021. KJW Health all rights reserved.

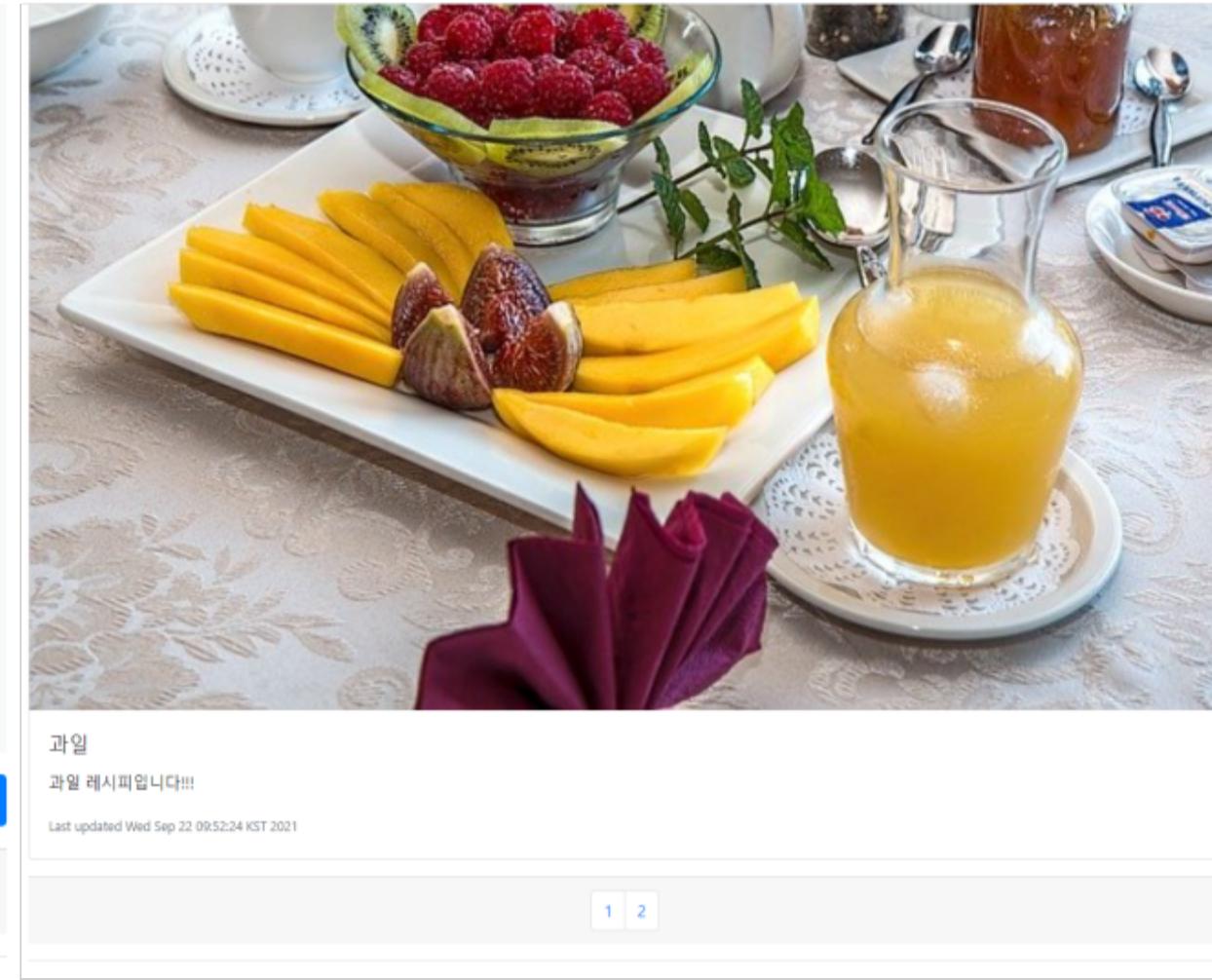
Plan

뉴스 페이지 또한 뉴스 기사에 추천 기능을 추가하여 추천수를 기준으로 인기 기사를 설정하여 기사를 통해 좀 더 유용한 정보를 얻어갈 수 있으면 좋을 것 같습니다.

9 KJW Health PROJECT 건강용품/식품, 식단레시피 화면 / 메인페이지

The screenshot shows the main homepage of the KJW Health project. At the top, there's a navigation bar with the KJW Health logo, a search bar, and a login link. Below the header, there are sections for 'Home', '진단하기' (Diagnosis), '건강용품/식품' (Health Products/Food), '건강 정보 공유' (Health Information Sharing), and '나의 건강 기록' (My Health Record). The '건강용품/식품' section features cards for '자전거' (Bicycle), '저항 밴드' (Resistance Band), '단백질 과자' (Protein Snack), '고구마' (Sweet Potato), '토마토' (Tomato), and '줄넘기' (Jump Rope). To the right, there's a '건강 공유 게시판' (Healthy Sharing Board) with posts like '김진우님 환영합니다!!' (Welcome Kim Jin-woo!) and '마이 페이지' (My Page). A sidebar on the right lists various numbers (131313, 121212, etc.) and a '물품 등록하기' (Register Product) button.

This screenshot displays a grid of cards for various health products and recipes. The cards include '고구마' (Sweet Potato), '토마토' (Tomato), '줄넘기' (Jump Rope), '샐러드' (Salad), '케틀벨' (Kettlebell), and '영양제' (Supplements). Each card has a small image, the name in Korean, and a blue '물품 등록하기' (Register Product) button at the bottom right. Navigation buttons '1' and '2' are visible at the bottom of the grid.



Introduce

- 메인페이지

마크업 언어를 이용하여 꾸며놓은 메인페이지입니다.

화면에 보이는 자료들을 DB에 저장된 정보를 가져와 꾸며보았습니다.

- 식단 레시피 페이지

DB에 저장된 식단을 모두 가져와서 페이징 기능을 추가하여 구현하였습니다.

- 건강용품/식품 페이지

마찬가지로 DB에 저장된 식단을 모두 가져와서 페이징 기능을 추가하여 구현하였습니다.

Plan

기회가 된다면 Front기술을 공부하여 조금 더 웹페이지 답게 꾸며보고 싶습니다.

식단 레시피 페이지에서 식단을 클릭하였을때 , 그 식단의 정보(레시피 등등)를 볼 수 있는 상세페이지 구현

DB설계를 더 구체적으로 하여 더 다양한 서비스를 이용할 수 있도록 구현해보고 싶습니다.

10 KJW Health PROJECT 마이페이지 / 관리자 페이지

나의 최근 5개 게시물
— KJW Health My Post

[내가 쓴 게시물 모두 보러가기](#)

No.	제목	작성자	시간
40	131313	1	2021-10-14 00:35:30
39	글쓰기 기능 테스트!!!	1	2021-10-13 23:04:08
31	121212	1	2021-10-10 00:00:22
30	111111	1	2021-10-09 21:04:24
29	101010	1	2021-10-09 21:04:18

나의 최근 3개 진단결과

— KJW Health My DiagnosisResult

[내가 기록한 진단결과 모두 보러가기](#)

기록된 Date : 2021-10-13 23:15:59

[자세히 보기 ▾](#)

기록된 Date : 2021-10-10 11:48:51

[자세히 보기 ▾](#)

기록된 Date : 2021-10-08 15:24:06

[자세히 보기 ▾](#)



관리자님 [로그아웃](#)

건강용품/식품 입력

minBMI :

maxBMI :

이름 :

[파일 선택](#) 선택된 파일 없음

[홈으로](#)

[저장](#)

Copyright 2021. KJW Health all rights reserved.

Introduce

- 마이페이지

내가 쓴 게시글 및 내가 기록한 진단결과 기록들을 한눈에 볼 수 있게 화면 구성을 해두었습니다.

- 관리자 페이지

데이터베이스에 저장된 정보를 진달결과에 BMI수치를 기준으로 추천해주게 됩니다.

데이터베이스에 저장하는 과정이 관리자 페이지에서 이루어집니다.

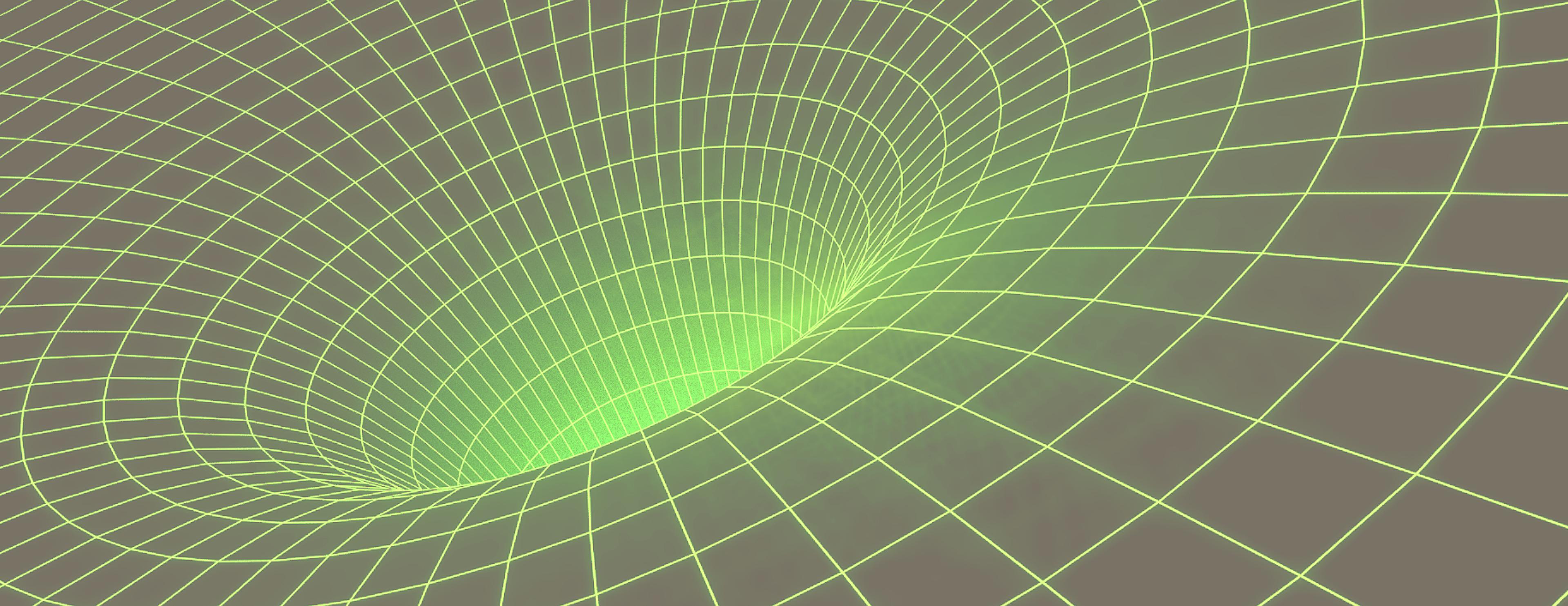
관리자로 설정하고 싶은 아이디를 관리자로 설정해두고, 관리자로 설정된 아이디로 로그인

했을 때 우측 상단 부분에 세팅 아이콘을 배치해두어 관리자페이지로 넘어갈 수 있도록

구현해두었습니다. 모든 자가진단결과 / 식단 / 건강용품식품 INSERT는 관리자페이지에서 처리됩니다.

Plan

계정별로 등급을 지정하는 기능(유저별로 활동을 얼마나 했는지에 따라서) 추가하여 마이페이지 / 게시판에서 개인의 계정 등급이 보여질 수 있도록 구현



감사합니다.